Matematik Bölümü

ENF 204 Bilgisayar Programlama

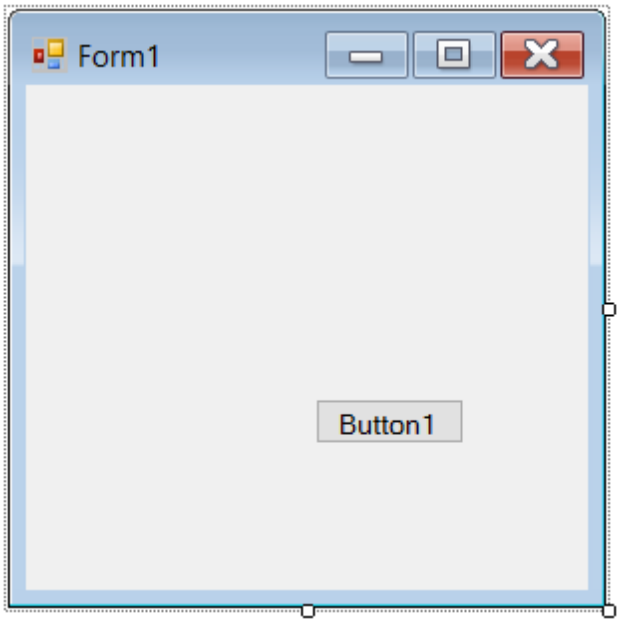
Ders Uygulamaları

Dr. İrfan MACİT

Adana, 2017

Genel Giriş ve Tanımlamalar

Visual Basic bilgisayar programlama görsel ve olay sürüşü (event driven) yöntemli programlama dilidir. Görsel olması kullanılan bileşenlerin (nesne) bir form (Form1) üzerinde yer almasından kaynaklanmaktadır.



Şekil 1. Form Ekranı.

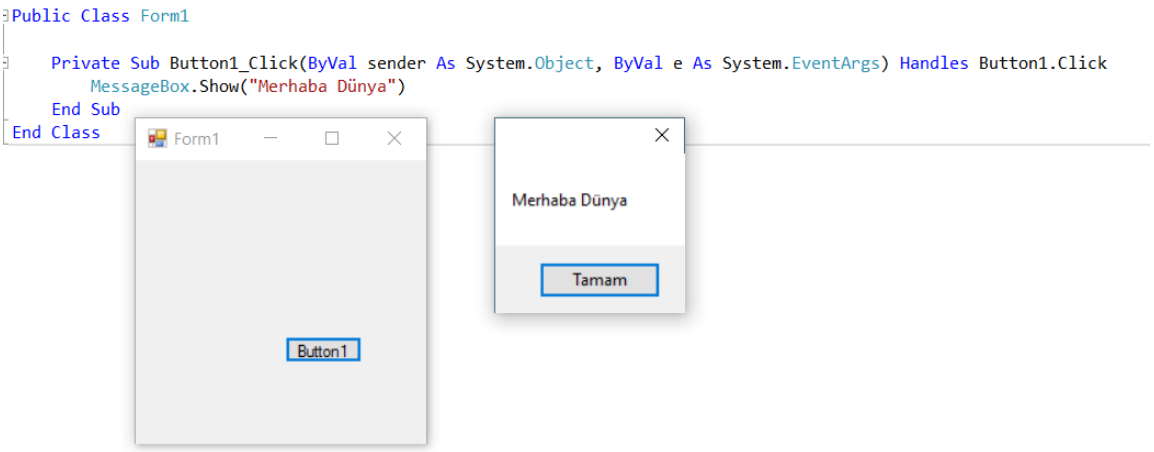
Bilindiği gibi bilgisayar programlama dillerinde veri giriş ve çıkışı belirli yöntemler kullanılarak yapılmaktadır. Veri tanımlanmadan kullanılmayan ifadelerdir. Bilgisayar programına veri göndermek, veriyi dış ortama aktarmak veya bilgisayar program kodları içerisinde işlem yapmak için kullanılan ifadelere değişken denir. Değişkenler içerdikleri veri türüne göre tipleri belirlenir. Örneğin değişkenin içerisinde tam sayı türünde bir ver işlenecek ise veri tipi tam sayı (integer) olmalıdır. Eğer değişken bir yazı veya harflerden oluşmuş ise string veya text denilen türlerde olmalıdır. Değişkenler tanımlanmadan kullanılamazlar. Bu yüzden bir değişken kullanılmadan önce mutlaka kullanılacağı türe göre tanımlanmalıdır. Kullanım sırasında değişken türü değiştirilemez, bunun tek istisnası kullanım sırasında kısmen dönüşüm yapmaktır. Kısmen dönüşüm yapılamayacak durum söz konusu olduğunda ise değişkenin tipinin tamamen yeniden tanımlanması gereklidir.

Visual basic programlama dili olay sürüşlü (event driven) olduğundan bahsetmiştik. Nesneler (bileşen, object) üzerinde Mouse ile tıklamak, gezmek veya seçmek bir olay oluşturur. Bu olaya göre yapılmak istenen kaynak kodları uygun alana yazılırlar. Uygun alanlar ise genellikle olayların tetiklendiği yöntemler (subroutine) satırları arasına yazılır. Bu yöntemlere altprogram (subroutine) de denir. Alt programlar **Sub** ifadesi ve **End Sub** ifadeleri arasındadır. Çalışmasını istediğimiz programı kodlarını Button1 nesnesi üzerine fare ile çift tıkladığımızda açılan pencere içerisindeki kod aralığına yazarız.

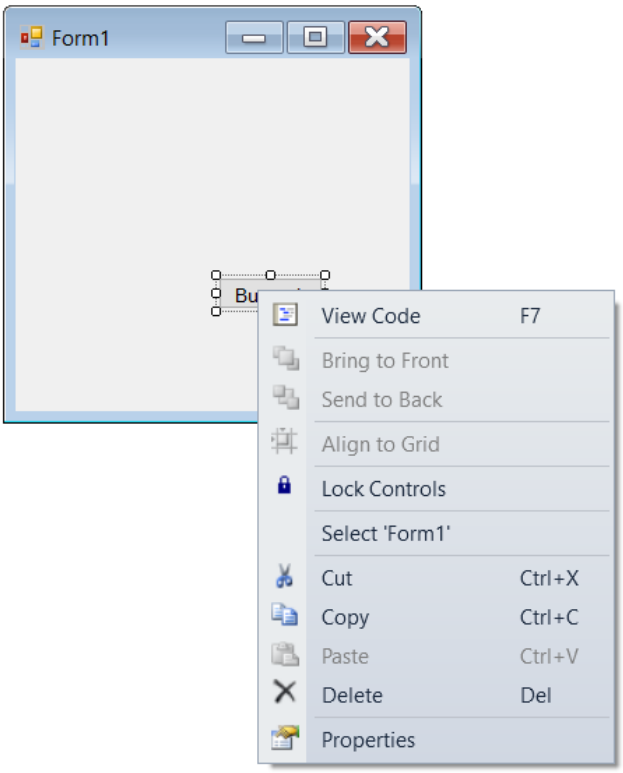
```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

    End Sub
End Class
```

Bütün bilgisayar programlar dillerinde ilk yazılan kaynak kodu “Merhaba Dünya” mesajıdır. Bu yazan mesajı ekranda göstermek için çift tırnak içerisine görünmesi istenen mesaj yazılır. Bu mesajı ekranda göstermek istersek *MessageBox* yöntemini kullanırız. Mesaj bilgisayar programından dış ortama bir veri göndermek için kullanılır. Burada unutulmaması gereken bir nesne çağrıldığında hangi özelliği kullanılacak ise bu özellik nokta ifadesinden sonra gelmelidir. Dış ortama veri aktaracak yöntem MessageBox ve özelliği ise Show olacaktır. Son olarak eğer tam olarak bu kodu yazmak istiyorsak program kaynak kodları aşağıdaki gibi olmalıdır.

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        MessageBox.Show("Merhaba Dünya")
    End Sub
End Class
```

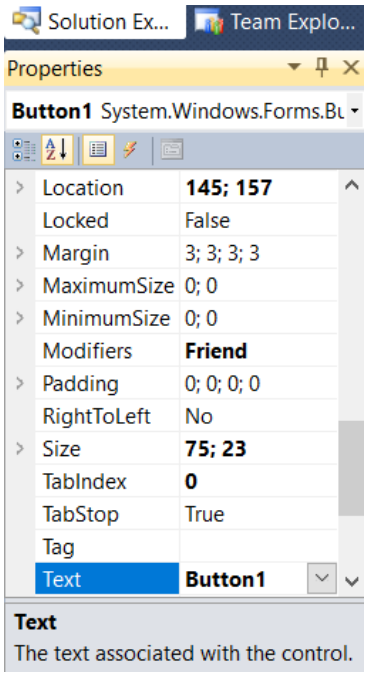


Çalışmasını istediğimiz kaynak kodları Button1 nesnesinin etiketini değiştirerek ekran da daha düzgün bir ifade görünmesini sağlayabiliriz. Bunun için Button1 nesnesinin Text özelliğini Göster olarak değiştirelim.



Şekil 2. Properties (Özellikler) Ekranı.

Değiştirme işlemine başlamadan önce Button1 nesnesinin üzerine fare ile giderek farenin sağ tuşuna basalım Şekil 2 ve ardından Properties (özellikler) Şekil 3'deki gibi seçelim.



Şekil 3. Properties (Özellikler) Değiştirme Diyalog Kutusu.

Düğmenin görünen etiketi için Text özelliği alanına Göster yazdığımızda Button1 düğmesinin görünen hali Şekil 4'teki gibi olacaktır.

Şekil 4. Button1 Nesnesinin Adının Değiştirilmesi.

Değişken Tanımlama

Buraya kadar bilgisayar programından dinamik veri çıkışını gösteren yöntemlerden sadece birisini kullandık. Bilgisayar programına dinamik veri girişi için kullanılan bir yöntem olan *InputBox* yöntemini kullanan örnek üzerinde duralım. Yine aynı kaynak kodlar üzerinden örneğe devam edilirse kaynak kodları aşağıdaki gibi değiştirilmelidir.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        InputBox("Bir Veri Girin", "Veri Girişi")
    End Sub
End Class
```

Buradaki kaynak kodlarda bilgisayar programına sadece dinamik veri girişinin nasıl yapıldığı gösterilmektedir. Girilen bu verinin kullanabilmek için değişken tanımlaması denilen yöntem ile girilen verinin bir değişkene atanması gerekir. Değişken tanımlamaları İngilizce'de Dimension kelimesinin kısaltılmışı olan *Dim* ifadesi kullanılır. Değişken tanımlama tam olarak *Dim as değişken_adı değişken_tipi* şeklindedir. Bu ifadelerin ne olduğunu kısaca açıklayalım. Değişken adı bilgisayar programında kullanılacak veya çağrılacak olan ifadedir. Değişken tipi ise değişkene aktarılacak olan değerin hangi türde olacağını göstermektedir. Değişkene aktarılacak değer aralığı hangi türde olacak ise o değişken tipi seçilmelidir.

Değişken veri tipleri için genel veri tipleri; tamsayılar için integer, ondalıklı sayılar için double, yazılar için string olmalıdır. Değişkenleri tanımlanmasında bazı kurallar vardır. Bu kurallar ;

i) Değişken adlarında Türkçe karakterler (İ, ı, ç, ö, ğ, ü, ş) kullanılmamalıdır,

ii) Değişkenler sayı ile başlayamazlar,

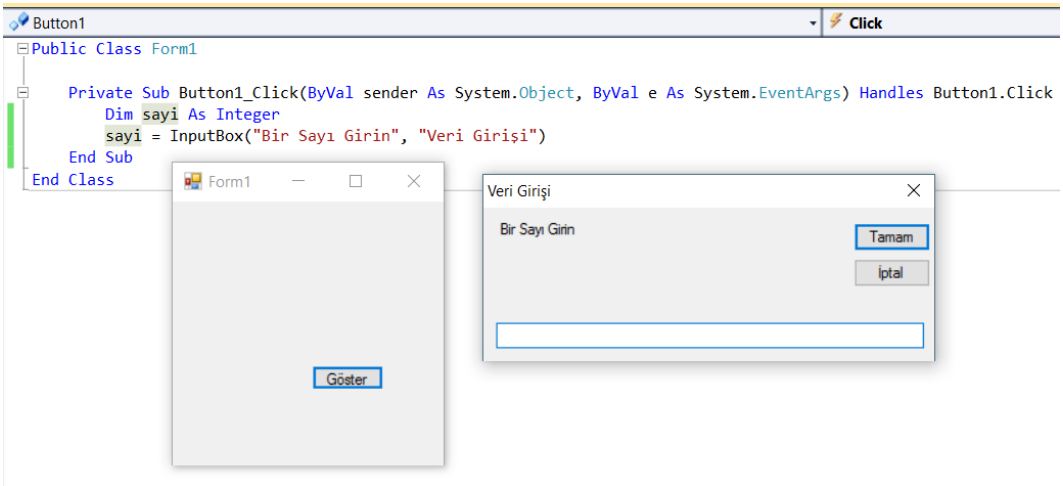
iii) Değişkenlerin adları iki ifadeden oluşacak ise aralarında boşluk olamaz,

iv) Altçizgi hariç herhangi bir operatörveya karakter olamaz (!,& , ( ), =).

Bu kurallara göre sayi (Dikkat "ı" kullanmıyoruz "i" olacak) adlı bir değişken tanımlayalım ve değişkene dış ortamdan bir değer girilmesini sağlayalım.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi As Integer
        sayi = InputBox("Bir Sayı Girin", "Veri Girişi")
    End Sub
End Class
```

Program çalıştırıldığında ekrana veri girişi için Şekil 5'teki gibi bir görüntü gelecektir. Burada unutulmaması gereken bir diğer nokta ise *Button1* ifadesinin yanında bulunan *Click* özelliğidir. Bu özellik ile fare *Button1* üzerinde tıklandığında olay (event) gerçekleşecek ve kodlar çalışmaya başlayacak.

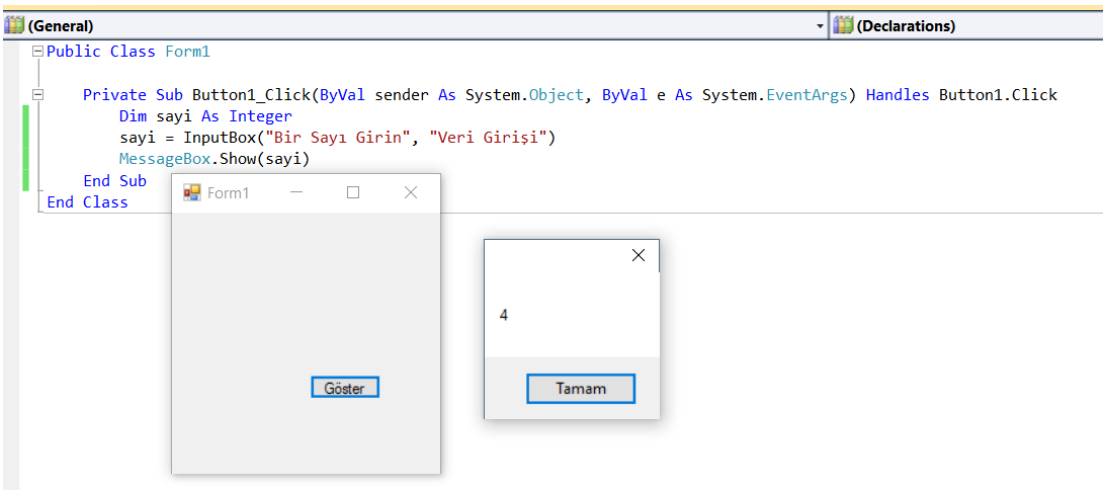


Şekil 5. Veri Giriş Ekranı.

Buraya kadar olan program kaynak kodları ile bilgisayar programına bir veri girişini inceledik. Girilen verinin ekranda görülmesini istenirse bir veri çıkışı yöntemi kullanılması gerekir. Girilen veri dinamik bir yöntem ile ekranda gösterilmek istendiğinde *MessageBox.Show* kullanılır. Bu yöntem bilindiği gibi

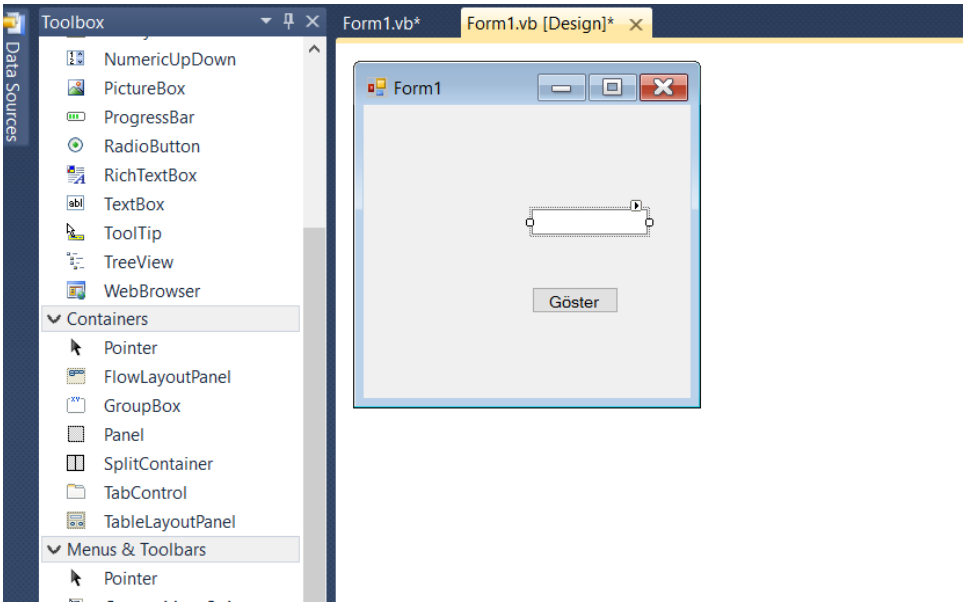
dinamik veri çıkış yöntemidir. Aşağıdaki kaynak kodlara göre klavyeden girilen sayi adlı değişkene 4 değerinde bir sayı girildiğinde ekranda Şekil 6'daki gibi görülecektir.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi As Integer
        sayi = InputBox("Bir Sayı Girin", "Veri Girişi")
        MessageBox.Show(sayi)
    End Sub
End Class
```



Şekil 6. Girilen Değişken Değerinin Ekranda Görüntülenmesi

Ekranda görünen Form1 nesnesi üzerinde veri girişi yöntemi olarak *TextBox.Text* yöntemi kullanılır. Burada *TextBox* yöntemi form1 nesnesi üzerine sol tarafta bulunan *Toolbox* kutusunun içerisinde seçilerek Form1 nesnesi üzerine bırakılır Şekil 7.



Şekil 7. Form1. Nesnesi Üzerinden Veri Girişi.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi As Integer
        sayi = TextBox1.Text
        MessageBox.Show(sayi)
    End Sub
End Class
```

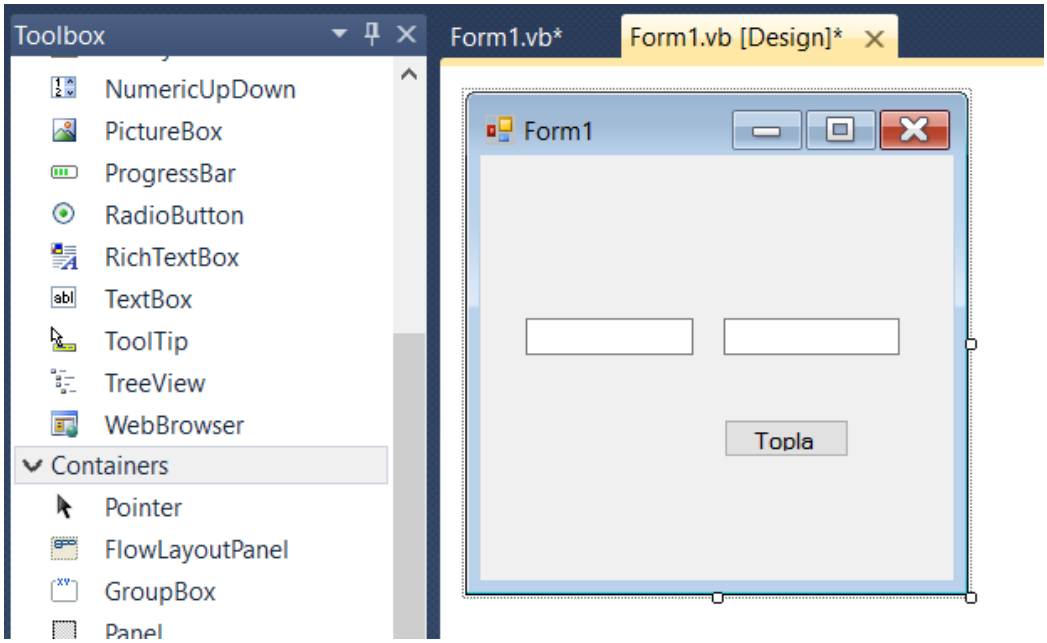
Kaynak kodları yenide düzenlendiğinde ise **`sayi = TextBox1.Text`** ile programa veri girişi yapılabilir. Program çalıştırıldığında *Form1* nesnesi üzerinde bulunan *TextBox1.Text* ile girilen 4 değeri ekranda Şekil 6'daki gibi görüntü gelecektir.

Visual Basic programlama dili *Form1* üzerinden girilen değerleri yazı tipinde kabul eder. Bu veri girişi dikkat edilmesi gereken bir durumu gerektirir. Eğer *Form1* üzerinden *TextBox1* gibi bir yöntem kullanarak sayısal bir değer girmek istiyorsak bu girilen değişkeni **Val()** fonksiyonu ile sayısallaştırmamız gerekir. Bu durum iki veya daha fazla sayısal değerin aritmetiksel işlemlerde kullanılması gerektiğinde önemli bir hal alacaktır. Eğer iki sayıyı *TextBox1.Text* yöntemi ile aritmetiksel işleme tutacaksak **Val()** fonksiyonuna ihtiyacımız olacaktır. Bu fonksiyonun kullanımı aşağıdaki gibidir.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi As Integer
        sayi = Val(TextBox1.Text)
        MessageBox.Show(sayi)
    End Sub
End Class
```

İki sayıyı toplamak için Form1 nesnesi üzerinden veri giriş yöntemini kullanarak toplama işlemini yapan program kaynak kodları aşağıdaki gibi olacaktır. Programın tasarımı Şekil 8'de görüldüğü gibi olacaktır. Burada ek olarak iki sayının tanımlanmasının yanı sıra toplamı alan bir sonuç değişkeni daha tanımlanmıştır. Sonuç değişkeni olan toplam adlı değişken ekranda sonucu gösteren bir değişkendir.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi1, sayi2, toplam As Integer
        sayi1 = Val(TextBox1.Text)
        sayi2 = Val(TextBox2.Text)
        toplam = sayi1 + sayi2
        MessageBox.Show(toplam)
    End Sub
End Class
```



Şekil 8. İki Sayının Toplamı From1 Nesnesi Üzerinde Veri Girişi.

Burada bir önemli bir konu toplam değişkeni tanımlanmadan program çalıştırılabilir. Fakat sonuç değişkenini tanımlanması programın çalışması ve kontrolünde önemli bir konudur.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi1, sayi2, toplam As Integer
        sayi1 = Val(TextBox1.Text)
        sayi2 = Val(TextBox2.Text)
        MessageBox.Show(sayi1 + sayi2)
    End Sub
End Class
```

Toplam değişkeni tanımlanmadan ekranda *MessageBox.Show* yöntemi ile iki sayının aritmetik operatör yardımı ile toplamı alınabilir. Bu çok tercih edilen bir durum değildir. Sonuç değişkenlerinin tanımlanması konusunda bir önemli konu da operatör işlemleri sonucunda alacağı değerin tipinin belirlenmesidir. Kısaca açıklamak gerekirse operatör işlemleri sonucunda verinin tipi değişebilir.

İki tam sayının toplamı tam sayı olabilir fakat iki tam sayının bölümü tam sayı olmayabilir. Bu durumda iki tam sayının bölümü gerçel (Double) sayı olabileceğinden sonuç değişkenin alacağı veri tipinin doğru tanımlanması gereklidir. Buna göre iki sayının bölümünü gösteren programın kaynak kodlar aşağıdaki gibi olacaktır.

```
Public Class Form1
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi1, sayi2 As Integer
        Dim bolum As Double
        sayi1 = Val(TextBox1.Text)
        sayi2 = Val(TextBox2.Text)
        bolum = sayi1 / sayi2
        MessageBox.Show(bolum)
    End Sub
End Class
```

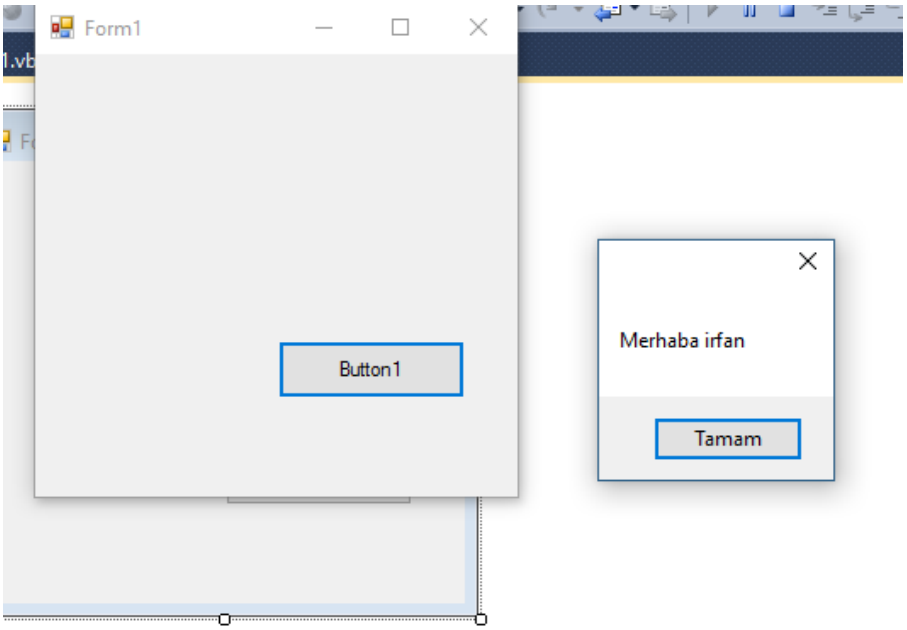
Toplamları alınacak olan sayılara veri girişini dinamik olarak yapmak istenirse burada *TextBox1.Text* ve *TextBox2.Text* yöntemi yerine *InputBox* yöntemi kullanılmalıdır.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sayi1, sayi2 As Integer
        Dim bolum As Double
        sayi1 = InputBox("İlk Sayıyı Girin", "Veri Giriş Kutusu")
        sayi2 = InputBox("İkinci Sayıyı Girin", "Veri Giriş Kutusu")
        bolum = sayi1 / sayi2
        MessageBox.Show(bolum)
    End Sub
End Class
```

Programlama dillerinde veri işlemleri oldukça önemlidir. Bu veri işleme yöntemlerinde değişkenleri tanımlanması ve kullanılması sırasında bazı özel durumlardan bahsedilebilir. Değişkenlerin veri giriş ve çıkışında formatlanması, bir başka değişkenin yanında yazılması veya bir mesaj ile birlikte yazılması gerekebilir. Böyle bir durumda İngilizce dilinde "ve" ifadesine karşılık gelen değişken birleştirme operatörü (&) karakterine ihtiyaç duyulur. Girilen değişkenin ekranda gösterilmesi sırasında mesajın yanında değişken yazılması için aşağıdaki program kaynak kodları yazılır.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim adi As String
        adi = InputBox("Adınızı Girin", "Veri Girişi")
        MessageBox.Show("Merhaba " & adi)
    End Sub
End Class
```

Kaynak kodları çalıştırdığımızda gelen *InputBox* veri giriş kutusundan gireceğimiz adi değişkeninin değer *MessageBox.Show* ile ekranda Şekil 9'daki gibi görünecektir.



Şekil 9. Girilen Verinin Ekranda Mesaj ile Gösterilmesi

Program tasarım ekranında *Form1* nesnesinin başlık değerinin değiştirilmesi için gerekli kaynak kodları aşağıdaki gibi olacaktır. Burada program çalıştırıldığında *Button1* nesnesi çağrıldığında *Form1* nesnesinin başlığı değişir.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.Text = "Bilgisayar Programım"
    End Sub
End Class
```

Program çalıştırılması sırasında Form1 nesnesinin başlığı girilen değer ile değiştirilmek isteniyorsa veri girişi için kullanılan bir yöntem ile alınan değer *Me.Text* değerine aktarılır. Bu işleme ait program kaynak kodları aşağıdaki olacaktır.

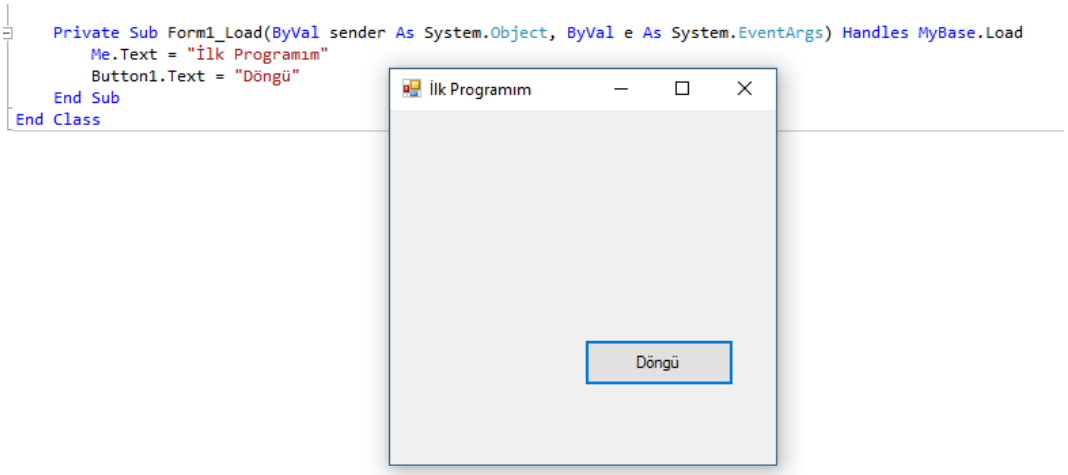
```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim baslik As String
        baslik = InputBox("Başlık Girin", "Veri Girişi")
        Me.Text = baslik
    End Sub
End Class
```

Tanımlanan *baslik* değişkeni ile *InputBox*'tan alınan değişken değeri *Me.Text* değişkenine aktarılmaktadır.

Program çalıştırılmaya başlandığında başlangıç ayarları veya başlangıçta beklenenler varsa bunları Form_Load içerisine yapmamız beklenir. Bunun için program tasarım ekranında Form1 nesnesi üzerine çift tıklama yapılır ve açılan program kodları arasına başlangıçta çalışmasını istediğimiz kodlar yazılır.

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Me.Text = "İlk Programım"
        Button1.Text = "Döngü"
    End Sub
```

Bu program kodları program çalıştırma düğmesine bastığımızda otomatik olarak programımıza yüklenecek ve programa ait *Form1* ve *Button1* nesnelerinin etiketlerini gösteren *Text* özellikleri Şekil 11'da görüldüğü gibi değiştirilecektir.



Şekil 11. Başlangıç Ayarları

Döngüler

Programlamada birden fazla tekrarlanacak olan işlemleri gerçekleştirmek için kullanılan yöntemdir. Genel olarak tekrarlı yapılar olarak bilinen döngüler ikiye ayrılmaktadır.

i. Şart olmayan döngüler,

ii. Şartlı döngüler.

Şartlı olmayan döngüler çalıştırılmadan önce kaç kez döngünün tekrarlanacağı bellidir. Şarta bağlı döngülerde ise durum farklıdır. Şartlı döngülerde tekrarlanacak olan kodların kaç kez tekrarlanacağı

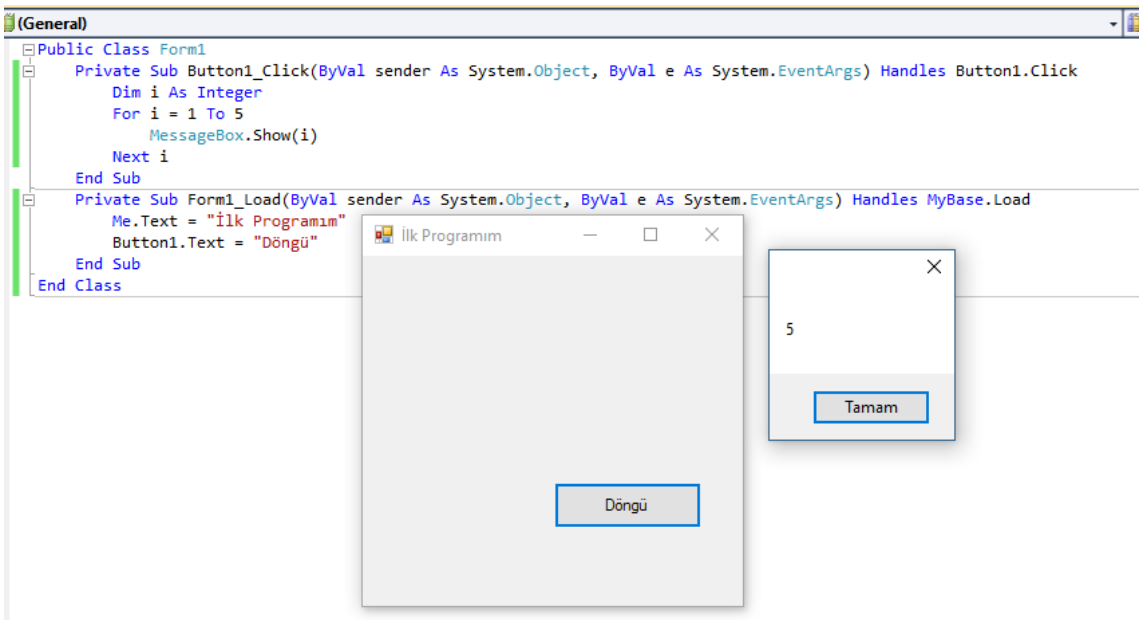
belirli değildir. Şart sağlandığı sürece veya şart sağlanana kadar kodlar çalıştırılacağına şart ne zaman sağlanırsa program kodlarının çalışması o zaman durur. Böylelikle daha önceden tahmin edilemeyen şart durumlarında programın döngüsünün kaç kez tekrarlanacağı bilinmez.

Şarta Bağlı Olmayan Döngüler

Döngülerde tekrarlama işlemi kontrol değişkeni tarafından kontrol edilir. Kontrol değişkeni tekrar edecek program kodlarının sayısını sayarak durma sayısını hesaplar. Döngüde kontrol değişkeni ile tekrar sayısı tutulduğundan kaç tekrar yapıldığı veya kaç tur atıldığı kontrol değişkenini saydırarak elde edilebilir. Genel şarta bağlı olamayan döngü yapılı *for/next* yapısıdır. Bu yapıda *for* ifadesi ile başlayan yapı **next** ifadesinden sonra kontrol değişkenini kullanılması ile sona erer.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i As String
        For i = 1 To 5
            MessageBox.Show(i)
        Next i
    End Sub
End Class
```

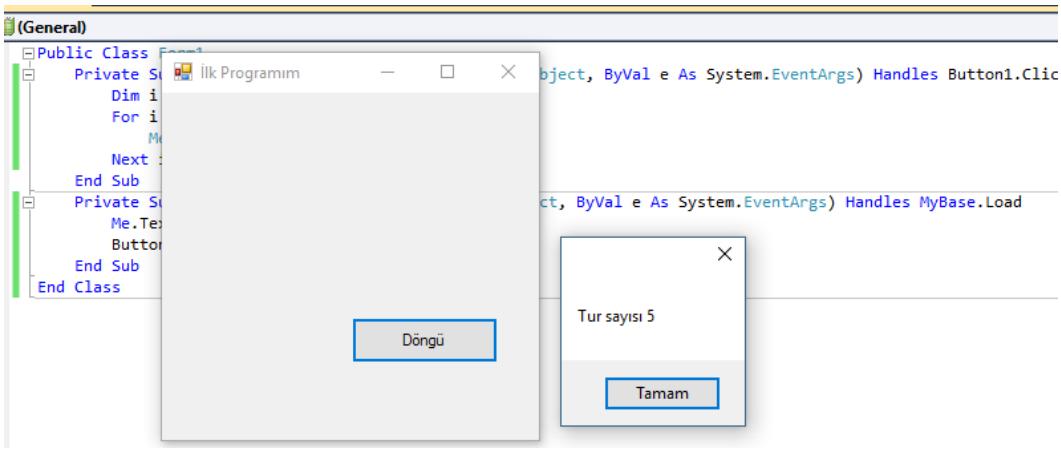
Basit olarak 1'den başlayarak 5'e kadar tekrarlayan *for* döngü yapısı *MessageBox.Show* yöntemi ile Şekil 12'de görüldüğü gibi sırası ile 1, 2, 3, 4 ve 5 sayılarının görülmesini sağlar.



Şekil 12. Şarta Bağlı Olmayan Döngü

Bu program kaynak kodlarında ekrana kaç tur atıldığını gösteren mesajı ekleyerek kaynak kodları yeniden düzenlediğimizde program aşağıdaki gibi olur Şekil 13.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i As Integer
        For i = 1 To 5
            MessageBox.Show("Tur sayısı " & i)
        Next i
    End Sub
```



Şekil 13. Ekranda Tur Sayısı Mesajı

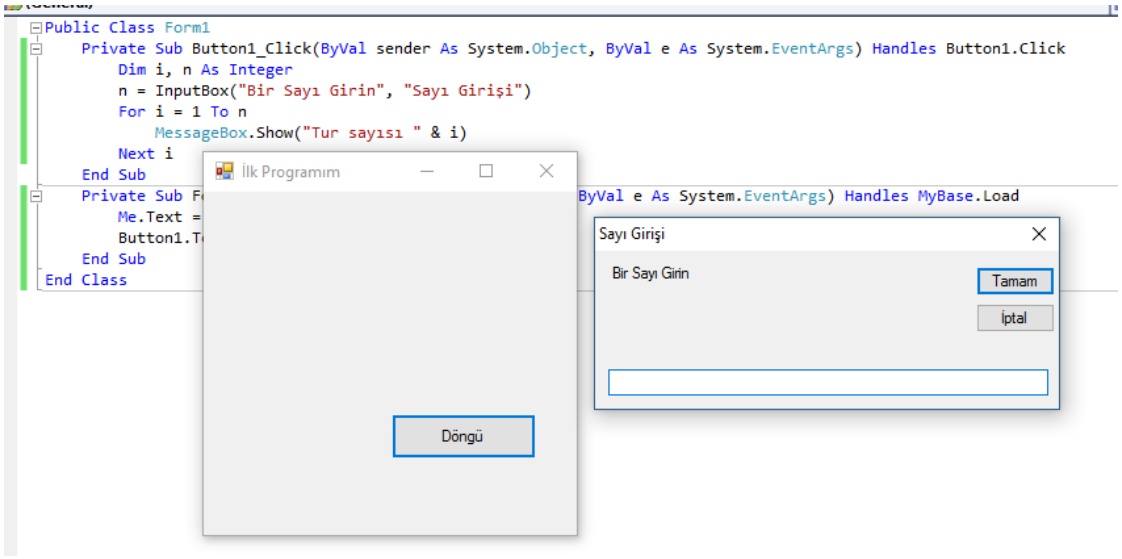
Şarta bağlı olamayan döngülerin başlangıç ve bitiş değerleri program çalıştırılması sırasında değiştirilebilir. Program çağrıldığında döngünün başlangıç ve bitiş değerleri klavyeden girilebilir. Bu yapıyı kullanarak 1'den 5'e kadar sayıların toplamını alan program kaynak kodları geliştirilebilir. Bunun için döngü içerisine birikimli toplamını alacak olan sonuç değişkeni olan toplam değişkenini tanımlamamız gerekir.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim i, toplam As Integer
        toplam = 0
        For i = 1 To 5
            toplam = toplam + i
        Next i
        MessageBox.Show("Serinin toplamı: " & toplam)
    End Sub
```

Buradaki kaynak kodlarda toplam = 0 ile birikimli toplam alınacak değeri toplamın etkisiz elemanı olan sıfır değerine eşitliyoruz. Eğer bu eşitlemeyi yapmazsak program her çalışmasında bir önceki toplam değerini yeni toplam değerine ekleyerek yanlış sonuç bulacaktır. Bu sebepten dolayı toplam birikimli algoritmalardan hatırlayacak olursan bu algoritmalarda sonuç alınacak olan değişken başlangıçta toplamda etkisiz eleman olan sıfır değerine eşitlenir.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i, n As Integer
        n = InputBox("Bir Sayı Girin", "Sayı Girişi")
        For i = 1 To n
            MessageBox.Show("Tur sayısı " & i)
        Next i
    End Sub
```

Döngünün *InputBox* yöntemi ile klavyeden girilen *n* sayısına kadar çalıştırılması sağlanır. Program her çağrıldığında klavyeden *n* sayısı girilmesi beklenir, girilen *n* sayısına kadar program tekrar yapmaya devam eder Şekil 14. Burada başlangıç sayısı 1'den başlar ve tur sayısı *n* değerinde girilen değere kadar devam eder. Bu programın bir önceki döngüden farkına gelince bundan önceki program kaynak kodunda 1'den başlayan ve 5'te biten tekrar sayısı vardı. Bu kaynak kodunda ise tekrar sayısı program çalışmaya başladığında belirlenmekte ve girilen bu sayı kadar tur yapmaktadır.



Şekil 14. Döngüye Bitiş Değerinin Girilmesi

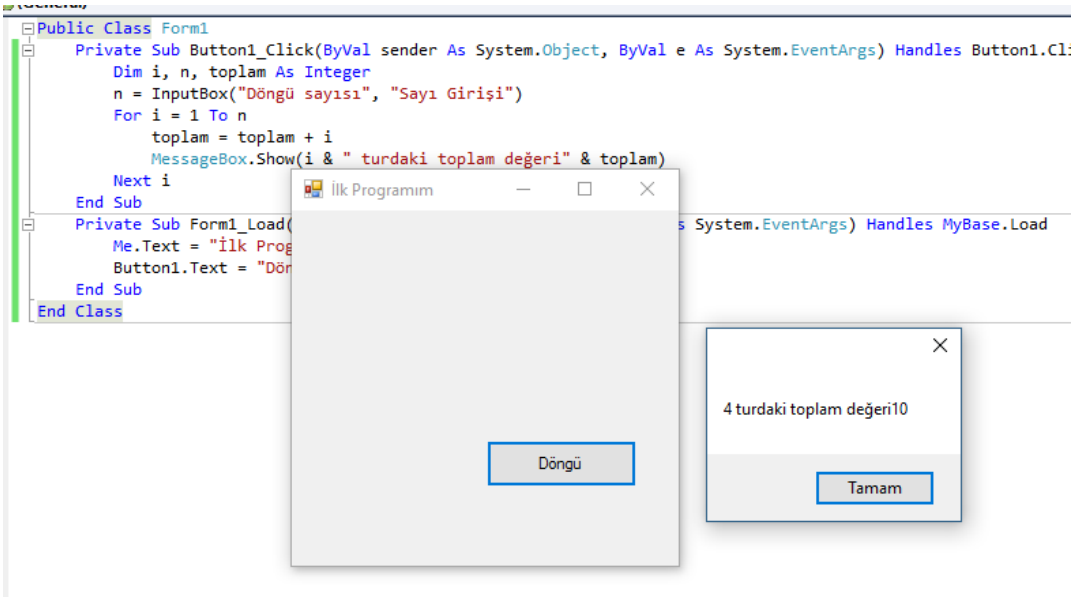
Burada döngünün bitiş değeri *InputBox* yöntemi ile girilen *n* değeri sayesinde belirlenmektedir. Bitiş değerine kadar tur sayısının belirlenen bir sayı aralığında artarak tamamlanması istersek bunun için *step k* (k: adım sayısını gösterir tamsayı) ifadesini kullanırız.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i, n As Integer
        n = InputBox("Döngü sayısı", "Sayı Girişi")
        For i = 1 To n Step 2
            MessageBox.Show(i)
        Next i
    End Sub
```

Döngüde kullanılan *step k* ifadesinde gösterilen tam sayı döngünün başlangıç sayısından itibaren adım aralığında artarak döngünün erişeceği n sayısına ulaşacağını göstermektedir. Toplam değerlerinin her turda ayrı olarak gösterildiği program kaynak kodları aşağıdaki gibi olacaktır.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i, n, toplam As Integer
        n = InputBox("Döngü sayısı", "Sayı Girişi")
        For i = 1 To n
            toplam = toplam + i
            MessageBox.Show(i & " turdaki toplam değeri" & toplam)
        Next i
    End Sub
```

Burada her turda birikimli toplam değerlerinin *MessageBox* yöntemi ile ekranda gösterilmesi sağlanır. Bazı durumlarda toplam değerinin her turda ayrı olarak ekranda gösterilmesi beklenir.



Şekil 15. Döngü Tur Toplam Değerinin Gösterilmesi

Ekrandaki gösterimde Şekil 15'te girilen *n* turun son değişken değeri 4 olduğunda ekranda tur ile birlikte son değerin de aldığı değer görülmektedir. Bu program kaynak kodunun devamında adım sayısını değiştirerek toplam almak istesek bunun için kaynak kodlarda aşağıdaki değişiklikleri yapmamız gerekecektir.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i, n, k, toplam As Integer
        n = InputBox("Döngü sayısı", "Sayı Girişi")
        k = InputBox("Adım sayısı", "Sayı Girişi")
        For i = 1 To n Step k
            toplam = toplam + i
            MessageBox.Show(i & " turdaki toplam değeri : " & toplam)
        Next i
```

```
        MessageBox.Show("Serinin toplam değerin: " & toplam)
    End Sub
```

Girilen *n* sayısına kadar seri toplamını alan programın kaynak kodlarında değişiklik yaparak klavyeden keyfi olarak girilen *5* adet sayının toplamını alan program kaynak kodları aşağıdaki gibi düzenlenebilir.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim i, n, toplam As Integer
        For i = 1 To n
            n = InputBox(i & ". sayıyı girin", "Sayı Girişi")
            toplam = toplam + n
        Next i
        MessageBox.Show("Girilen sayıların toplam değeri: " & toplam)
    End Sub
```

Program çalıştırıldığında klavyeden 5 adet sayı girmesi beklenir. Girilen bu ayrı sayıları birbirleri ile toplayarak sonuç toplam değerine ulaşır.