İNTERNET TABANLI PROGRAMLAMA- 3.ders

DİZİLER

Diziler bir çok bilgiyi tek bir değişken içerisinde tutmamızı sağlayan ifadelerdir. Dizide tutulan bilgiler Ram da tutulur. Elektrikler kesildiğinde dizideki bilgilerde kaybolacaktır. C# da dizi tanımlama iki şekilde olmaktadır. Bunlar boyutlu dizi tanımlama ve boyutsuz dizi tanımlamadır.

Bilgiler diziye her eklendiğinde dizinin sıfırlanmaması için tanımlamaları Globalde (alt yordamların en üstünde) yapmak gerekir. Dizideki ilk eleman her zaman [0] sıfır indisi ile tutulur. Dolayısı ile diziye bilgi eklerken yada okuturken ilk eleman sıfırıncı eleman olmalıdır.

Dizinin içinde kaç eleman olduğunu bilmiyorsak, dizideki eleman sayısınca döngünün dönmesini istiyorsak **foreach()** döngüsü kullanmak gerekir. Bu döngünün yapısı şu şekildedir.

Foreach Döngüsü

Bu döngü diziler için kullanımı kolay bir döngüdür. Eğer bir dizideki tüm elemanlar üzerinde işlem yapmak istiyorsak ve dizinin eleman sayısını bilmiyorsak kullanabiliriz. Döngü her döndüğünde diziden sırayla okunan eleman bir değişkene atılır ve döngü içinde de bu değişken kullanılır.

```
foreach (string Eleman in Dizi)
{
      listBox1.Items.Add(Eleman);
}
```

Boyutlu Dizi Tanımlama

Bu tanımlamada dizinin eleman sayısı ve tipi belirlenmelidir. Örnek tanımlama şekli aşağıdaki gibidir.

```
int [] Dizi = new int[100];
```

Dizideki elemanlar okunurken dizinin boyutu bilindiği için for() döngüsü kullanılabilir. Tabiki diziler için en kullanışlı döngü foreach() döngüsüdür. Bu döngüyü kullanmak daha çok tercih edilmelildir.

Boyutsuz Dizi Tanımlama (ArrayList Dizisi)

Bu dizide dizinin boyutu ve tipini belirlemeye gerek yoktur. Normalde dizilerde tüm elemanlar tanımlanan tipde olmak zorundadır. Fakat bu dizi tanımlamasında farklı tipleri (string, int vs) aynı dizi içerisinde tutmak mümkündür.

ArrayList komutu ile dizi tanımlayabilmek için aşağıdaki kütüphanenin sayfaya eklenmiş olması gerekir.

```
using System.Collections;
```

Dizinin tanımlaması aşağıdaki gibi yapılır.

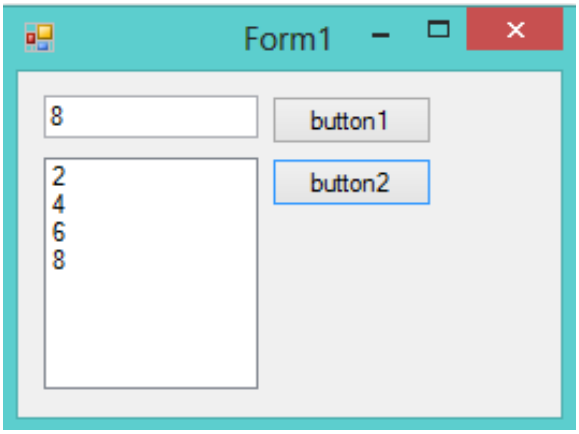
```
ArrayList Dizi = new ArrayList();
```

ArrayList ile ilgili olarak kullanabileceğimiz bazı komutlar şunlardır.

```
Dizi.Add(TextBox1.Text);//Diziye eleman ekler
Dizi[i].ToString();     //Diziden okumayı sağlar
Dizi.Clear();           //dizi içerisindeki tüm elemanları siler.
Dizi.Count;             //dizinin eleman sayısını verir.
Dizi.RemoveAt(3);       //indis numarası 3 olan elemanı Diziden siler.
```

Örnek

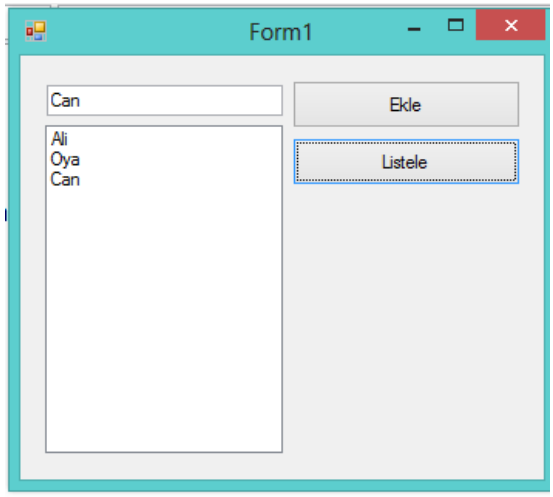
Şekildeki gibi bir Form üzerine 2 buton, 1 Texbox, 1 Listbox ekleyin. Textbox'a girilen sayıları birinci butona tıklayınca diziye eklesin. Daha sonra ikinci butona tıklandığında tüm bilgiler Diziden okunup ListBox a eklensin.




```
using System.Collections;
int [] Dizi1 = new int[100];
int i=0;

private void button1_Click(object sender, EventArgs e)
{
      i++;
      Dizi1[i] =Convert.ToInt32(textBox1.Text);
}

private void button2_Click(object sender, EventArgs e)
{
      foreach (string eleman in Dizi1)
      {
            try
            {
            listBox1.Items.Add(eleman);
            }
            catch { }
      }
}
```

Örnek

Yukarıdaki aynı örneği boyutsuz dizi (ArrayList) kullanarak yapın.

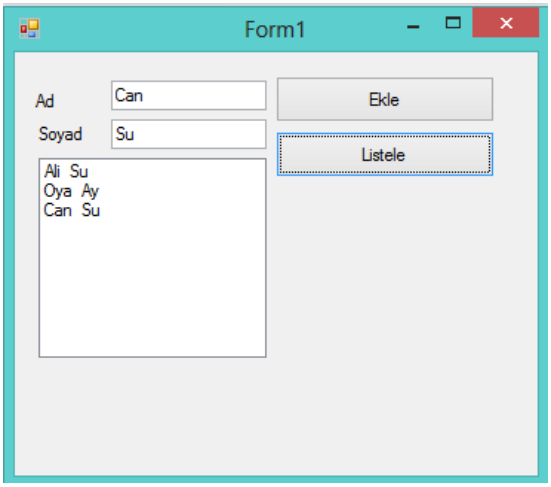



```
ArrayList Dizi = new ArrayList();

private void button1_Click(object sender, EventArgs e)
{
      Dizi.Add(textBox1.Text);
}

private void button2_Click(object sender, EventArgs e)
{
      foreach (string Eleman in Dizi)
      {
            listBox1.Items.Add(Eleman);
      }
}
```

Örnek

Şekildeki gibi bir 2 Textbox dan Kişilerin Ad ve Soyad bilgilerini alın. Ekle butonuna tıklandığında her seferinde bu bilgileri tek boyutlu bir diziye eklesin. Ardından Listele butonuna tıklandığında kişilerin Ad ve Soyadlarını ListBox'da görüntülesin.

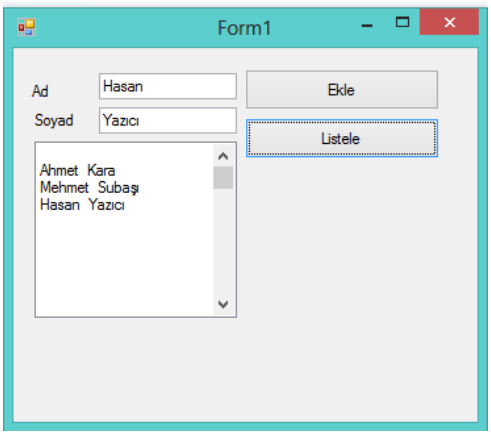
1. Yöntem




```
ArrayList DiziAd = new ArrayList();
ArrayList DiziSoyad = new ArrayList();

private void button1_Click(object sender, EventArgs e)
{
```

```
        DiziAd.Add(textBox1.Text);
        DiziSoyad.Add(textBox2.Text);

}

private void button2_Click(object sender, EventArgs e)
{
        for (int i = 0; i < DiziAd.Count; i++)
        {
                listBox1.Items.Add(DiziAd[i] + "   " + DiziSoyad[i]);
        }
}
```

2.Yöntem




```
string[,] Dizi = new string[100, 2];
int i = 0;

private void button1_Click(object sender, EventArgs e)
{
        i++;
        Dizi[i,0] = textBox1.Text; //Ad Eklendi
        Dizi[i,1] = textBox2.Text; //Soyad Eklendi

}
private void button2_Click(object sender, EventArgs e)
{
        for (int i = 0; i < 100 ; i++)
        {
                listBox1.Items.Add(Dizi[i,0] + "   " + Dizi[i,1] );
        }
}
```

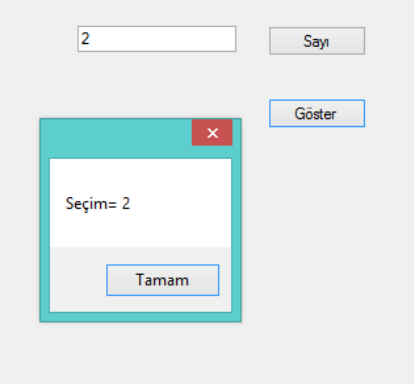
Switch Case Yapısı

Bu yapı If yapısı gibi koşullu durumlar için kullanılabilir. Burada koşulu sağlarken sadece eşit olma durumuna bakar. Büyük yada Küçük gibi koşulları kabul etmez. Değişken olarak kullanılacak ifadenin sadece integer olması gerekir. Yani blok yapıda kullanılacak değişkenini içeriği 1,2,3... gibi integer sayılar olmalıdır. Şu şekilde düşünebiliriz. Değişkenimizin içindeki değer 1 eşitse şu işlemi yap, 2 ye eşitse şu işlemi yap, gibi durumlar için kullanılır. Yapısı aşağıdaki şekildedir.

```
switch (Sayi)
 {
   case 1:
      .....
```

```
            break;

        case 2:
          .....
            break;

        case 3:
          ....
            break;

        default:
          .....
            break;

    }
```

Örnek:




```
int Sayi = 0;

private void button1_Click(object sender, EventArgs e)
{
   Sayi = Convert.ToInt32(textBox1.Text);
}

private void button2_Click(object sender, EventArgs e)
{
   switch (Sayi)
   {
      case 1:

          MessageBox.Show("Seçim= " + Sayi.ToString());
          break;

      case 2:

          MessageBox.Show("Seçim= " + Sayi.ToString());
          break;

      case 3:
```

```
            MessageBox.Show("Seçim= " + Sayi.ToString());
            break;

        default:

            MessageBox.Show("Geçersiz bir değer girdiniz" + Sayi.ToString());
            break;

    }

}
```

UYGULAMALAR

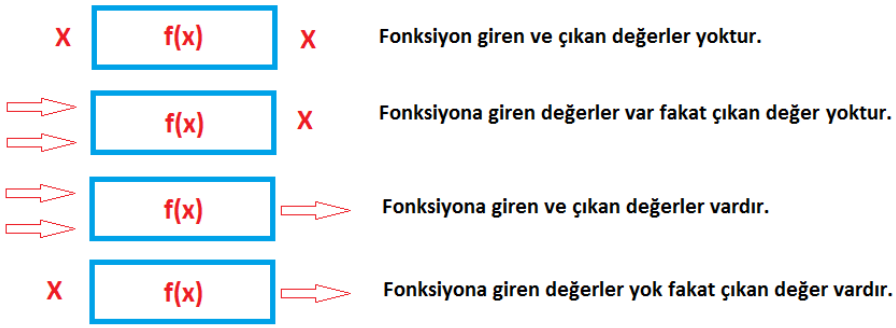
1. Dışırıdan bir buton ve textbox kullanarak girilen sayıları diziye ekleyin. Ardından başka bir butona tıklandığında bu sayıların içindeki asal sayıları listbox a ekleyen ve eklediği asal sayıların toplamını listenin altında veren programı yazınız.

FONKSİYONLAR

Program yazarken fonksiyon kullanmanın bir çok faydası vardır. Bu faydası aşağıdaki şekilde özetlenebilir.

a) Program fonksiyonlar vasıtasıyla daha küçük parçalara bölündüğü için programın anlaşılırlığı artar.
b) Fonksiyona yazılan komutlar programın değişik yerlerinde tekrar yazılmak zorunda kalınmadığı için program daha az kodla oluşturulmuş olur.
c) Fonksiyona giren ve çıkan değerler kontrol altında tutulduğu için, programda oluşabilecek hataların önüne geçilmiş olur ve hataların tesbiti kolaylaşmış olur.
d) Programın akış diyağramı ve mantıksal yapısı fonksiyon kullanımı ile daha kolay oluşturulur.
e) Nesne tabanlı programlama teknikleri kullanılırken, fonksiyon yapıları kullanılacağı için alt yapı sağlanmış olur.

Fonksiyonları aynı matematikte fonksiyonlar gibi düşünebiliriz. Matematik de bir fonksiyona bir çok değer girer, fonksiyon içerisinde bazı işlemler yapılır ve sonuç olarak da bir tane değer üretilir. $y= 3x^2 + z$ şeklinde verilen bir fonksiyonda x ve z değerleri giren değerler, y değeri ise çıkan değerdir. x ve z değerleri bazı işlemlerden geçilir ve tek bir çıkış olan y değeri oluşturulmuş olur. Aynı şekilde programlamadaki fonksiyon ifadeleri ise giriş ve çıkış değerlerine bağlı olarak 4 şekilde gruplandırılabilir.



Fonksiyonun genel formatı şu şekildedir. Fonksiyona giren değişken değerleri parantezin içinde tanımlanır. Dışarıdan bu fonksiyona değerler gönderilirken, girişteki tanımlanan değişkenin tipleri ile aynı tipte olmalıdır. Fonksiyondan geri değer döndürülecekse fonksiyon içinde return kelimesi ile geri dönen değer gösterilir. Bu değer fonksiyonun adının başında bulunan tip ile aynı olmalıdır. Eğer bir fonksiyon geri değer göndermeyecekse fonksiyonun adının başına void ifadesi eklenmeidir. Bu ifadeden önce public ifadesi kullanılırsa fonksiyon programın her tarafından çağrılarak kullanılabilir.

```
public double FonksiyonunAdi(double GirenDegisken1, double GirenDegisken2)
{
      double FonksiyonIcindeKullanilanYerelDegisken= 0;
      .......
      İşlemler
      .......
      return GeriDonenDeger;
}
```

Fonksiyon Oluşturma

Fonksiyonlar dört farklı tipte oluşturulur.

A- Hiç bir dış değer almayan ve geri değer göndermeyen fonksiyonlar. Sadece işlem yapar	B- Dışarıdan değer almaz fakat geri değer gönderir
public Void FonksiyonAdi() *{* *Yerel degişkenler*	*public geridönüştipi FonkAdi()* *{* *Yerel degişkenler*

İşlemler *}*	*İşlemler* *Return geridönendeğer* *}*

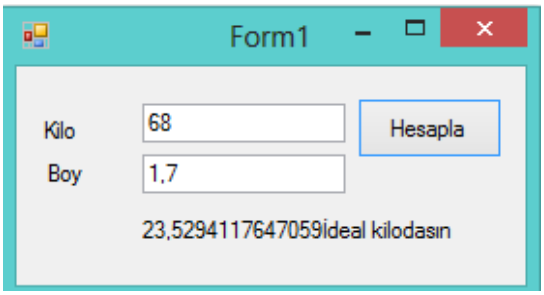
private void button1_Click(object sender, EventArgs e)
{
Hesapla();
}
 	
public void Hesapla()
{
int A=3, B=2, C;
C = A + B;
MessageBox.Show("Sonuc="+C);
}
	private void button1_Click(object sender, EventArgs e)
	{
	MessageBox.Show(Hesapla().ToString());
	}
	
	public int Hesapla()
	{
	int A=3, B=2, C;
	C = A + B;
	return C;
	}

C- Dışarıdan değer alır fakat dışarı değer göndermez	D- Dışarıdan hem değer alır hemde değer göderir.
public Void FonksiyonAdi(tip Degişken1, tip Degişken2) *{* *Yerel degişkenler* *İşlemler* *}*	*public geridönüştipi FonkAdi(tip Degişken1, tip Degişken2)* *{* *Yerel degişkenler* *İşlemler* *Return geridönendeğer* *}*

private void button1_Click(object sender, EventArgs e)
{
int X = 3, Y = 2;
Hesapla(X,Y);
}
 	
public void Hesapla(int A, int B)
{
int C;
C = A + B;
MessageBox.Show(C.ToString());
}
	private void button1_Click(object sender, EventArgs e)
	{
	int X = 3, Y = 2;
	MessageBox.Show(Hesapla(X, Y).ToString());
	}
	
	public int Hesapla(int A, int B)
	{
	int C;
	C = A + B;
	return C;
	}

Örnek

İdeal kilo hesabı yapan bir program yazın. Bu program üzerinde 4 tip fonksiyon kullanımını gösterin. İdel Kilo = Kilo / Boy^2 dir. Örnek : 68 kg/ 1.70 m = 23.52 çıkar. Bu sayı 18 den küçük ise kişi zayıf, 18-25 arasında ise ideal kiloda, 25-30 arasında ise hafif şişman ve 30 üzerinde çıkarsa obozite demektir.



1.fonksiyon kullanımı

```
private void button1_Click(object sender, EventArgs e)
{
            Hesapla();
}

public void Hesapla()
{
      double Boy=0, Kilo=0,Katsayi = 0;

      Boy=Convert.ToDouble (textBox2.Text)  ;
      Kilo = Convert.ToDouble (textBox1.Text);

      Katsayi = Kilo / (Boy * Boy);
      if (Katsayi < 18)
            label3.Text =Katsayi.ToString() + "Zayıfsın";
      else if(Katsayi >= 18 && Katsayi <= 25)
            label3.Text = Katsayi.ToString() + "İdeal kilodasın";
      else if (Katsayi > 25 && Katsayi < 30)
            label3.Text = Katsayi.ToString() + "Hafif Şişmansın";
      else if (Katsayi > 30)
      label3.Text = Katsayi.ToString() + "Şişmansın";
}
```

2. Tip Fonksiyon Kullanımı

```
private void button1_Click(object sender, EventArgs e)
{
      double Boy = 0, Kilo = 0, Katsayi = 0;

      Boy = Convert.ToDouble(textBox2.Text);
      Kilo = Convert.ToDouble(textBox1.Text);

      Hesapla(Boy,Kilo);
}

public void Hesapla(double Boy_degisken, double Kilo_degisken)
{
      double  Katsayi = 0;
      Katsayi = Kilo_degisken / (Boy_degisken * Boy_degisken);

      if (Katsayi < 18)
            label3.Text =Katsayi.ToString() + "Zayıfsın";
      else if(Katsayi >= 18 && Katsayi <= 25)
```

```
            label3.Text = Katsayi.ToString() + "İdeal kilodasın";
      else if (Katsayi > 25 && Katsayi < 30)
            label3.Text = Katsayi.ToString() + "Hafif Şişmansın";
      else if (Katsayi > 30)
            label3.Text = Katsayi.ToString() + "Şişmansın";
}
```

3. Tip Fonksiyon Kullanımı

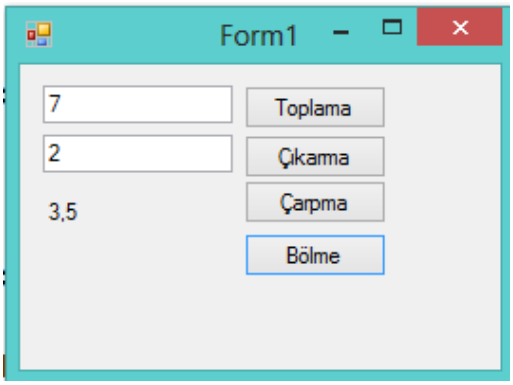
```
private void button1_Click(object sender, EventArgs e)
{
      double Boy = 0, Kilo = 0, Katsayi = 0;

      Boy = Convert.ToDouble(textBox2.Text);
      Kilo = Convert.ToDouble(textBox1.Text);

      double Katsayi_Degeri= Hesapla(Boy,Kilo);

      if (Katsayi_Degeri < 18)
            label3.Text = Katsayi_Degeri.ToString() + "Zayıfsın";
      else if (Katsayi_Degeri >= 18 && Katsayi_Degeri <= 25)
            label3.Text = Katsayi_Degeri.ToString() + "İdeal kilodasın";
      else if (Katsayi_Degeri > 25 && Katsayi_Degeri < 30)
            label3.Text = Katsayi_Degeri.ToString() + "Hafif Şişmansın";
      else if (Katsayi_Degeri > 30)
            label3.Text = Katsayi_Degeri.ToString() + "Şişmansın";
}

public double Hesapla(double Boy_degisken, double Kilo_degisken)
{
      double  Katsayi = 0;
      Katsayi = Kilo_degisken / (Boy_degisken * Boy_degisken);

      return Katsayi;
}
```

4. Tip Fonksiyon Kullanımı

```
private void button1_Click(object sender, EventArgs e)
{
      double Katsayi_Degeri = Hesapla();

      if (Katsayi_Degeri < 18)
            label3.Text = Katsayi_Degeri.ToString() + "Zayıfsın";
      else if (Katsayi_Degeri >= 18 && Katsayi_Degeri <= 25)
            label3.Text = Katsayi_Degeri.ToString() + "İdeal kilodasın";
      else if (Katsayi_Degeri > 25 && Katsayi_Degeri < 30)
            label3.Text = Katsayi_Degeri.ToString() + "Hafif Şişmansın";
      else if (Katsayi_Degeri > 30)
            label3.Text = Katsayi_Degeri.ToString() + "Şişmansın";
}

public double Hesapla()
{
      double Boy = 0, Kilo = 0, Katsayi = 0;

      Boy = Convert.ToDouble(textBox2.Text);
      Kilo = Convert.ToDouble(textBox1.Text);

      Katsayi = Kilo / (Boy * Boy);

      return Katsayi;
```

```
}
```

Örnek

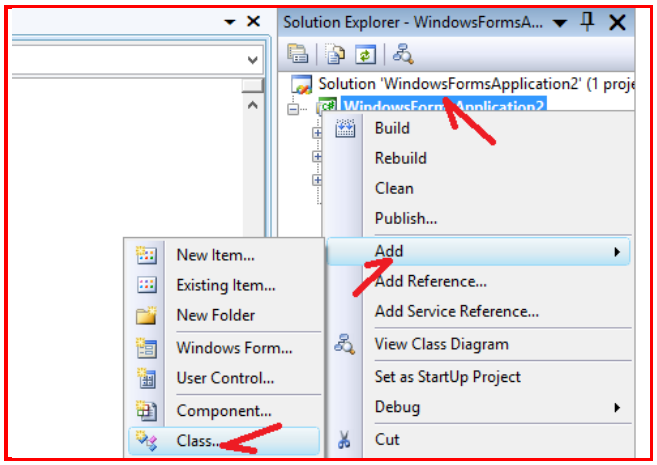


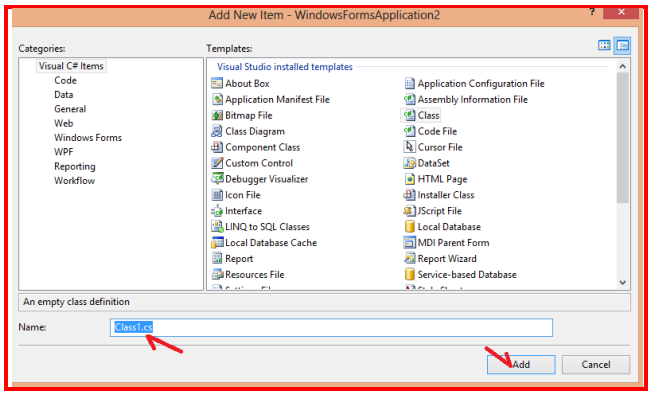

```
private void button1_Click(object sender, EventArgs e)
{
      topla();
}

private void button2_Click(object sender, EventArgs e)
{
      int Sayi1 = Convert.ToInt32(textBox1.Text);
      int Sayi2 = Convert.ToInt32(textBox2.Text);

      cikar(Sayi1, Sayi2);
}

private void button3_Click(object sender, EventArgs e)
{
      int Sayi1 = Convert.ToInt32(textBox1.Text);
      int Sayi2 = Convert.ToInt32(textBox2.Text);
      label1.Text = carpma(Sayi1, Sayi2).ToString();

}

private void button4_Click(object sender, EventArgs e)
{
      label1.Text = bolme().ToString();

}

//FONKSİYONLAR
public void topla()
{
      int Sayi1 = Convert.ToInt32(textBox1.Text);
      int Sayi2 = Convert.ToInt32(textBox2.Text);

      int sonuc = Sayi1 + Sayi2;
      label1.Text = sonuc.ToString();
}

public void cikar(int a, int b)
{
      int sonuc = a - b;
      label1.Text = sonuc.ToString();
}
```

```
public int carpma(int a, int b)
{
      return a * b;
}

public double bolme()
{
      int Sayi1 = Convert.ToInt32(textBox1.Text);
      int Sayi2 = Convert.ToInt32(textBox2.Text);

      double sonuc =(double) Sayi1 / (double) Sayi2;
      return  sonuc;
}
```

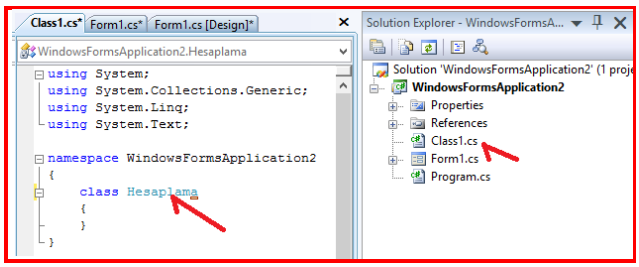
Sınıf (Nesne) Oluşturma

Projemizin içinde tekrar eden kodları bir nesne şeklinde (sınıf=class) şeklinde oluşturarak tekrar kullanabiliriz. Sınıf yapısını projenin her tarafından çağırıp kullanıma açabiliriz. Bunun için öncelikle solution penceresinden projemize sağ tuşa tıklarsak Add>Class yolunu takip edersek sınıf oluşturmak için pencere açılacaktır. Bu pencerede sınıf kodları için isim belirleyebiliriz.





Burada önemli olan Class1.cs isminden daha çok içerisindeki sınıf tanımlayan fonksiyonun adıdır. Bu ad bizim kodlarımızda kullanacağımız isimdir.



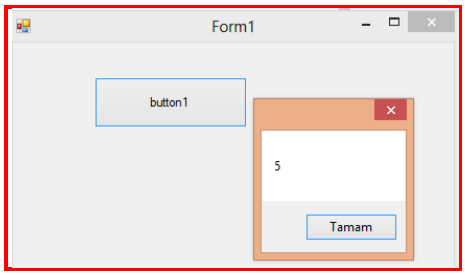
Şimdi bu hesaplama sınıfı içerisine iki tane fonksiyon (metod) yazalım.

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

namespace WindowsFormsApplication2
{
    class Hesaplama
    {
        public double Topla(double A, double B)
        {
            return A + B;
```

```
        }
        public double Cikar(double A, double B)
        {
            return A - B;
        }
    }
}
```

Sınıfı projemizde kullanırken şu şekilde tanımlarız.

```
private void button1_Click(object sender, EventArgs e)
{
Hesaplama Islem =new Hesaplama();
MessageBox.Show(Islem.Topla(2,3).ToString());
}
```



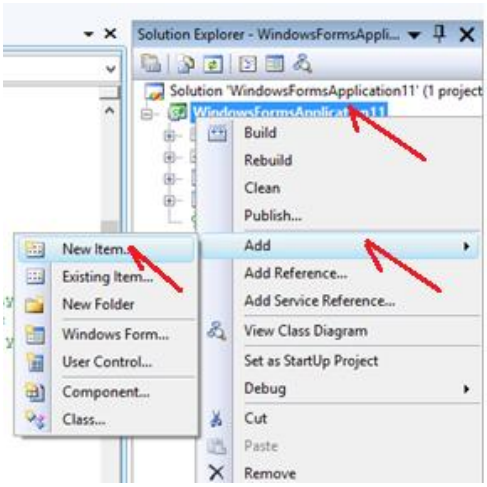
FORMLAR ARASI BİLGİ TAŞIMA

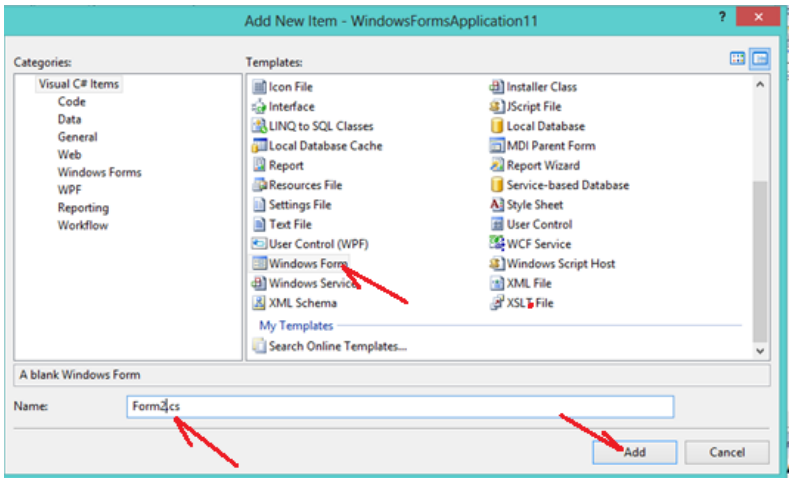
Masaüstü program hazırlarken projemizde bir çok form kullanabiliriz. Bu formlar geçişin nasıl olacağı ve önceki formlardaki bilgilerin diğer formlara nasıl taşınacağına dair aşağıdaki örneği inceleyin.

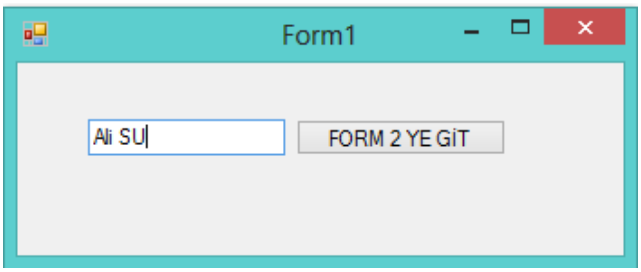
Örnek

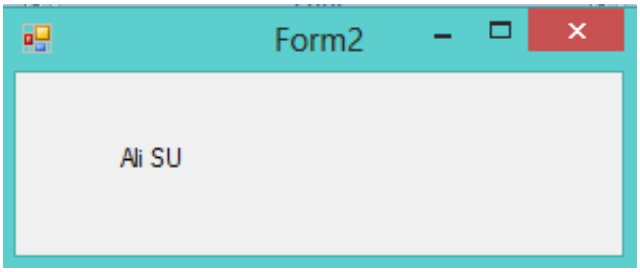
Form1 üzerine 1 buton ve textbox ekleyin. Texbox yazılan isim butona tıklanınca başka bir form açılsın ve orada bulunan labelda görüntülensin.

Öncelikle projemize yeni bir form eklememiz gerekir. Bunun için “Solution Explorer” dan proje başlığı üzerine sağ tuşa tıklayıp yeni form u ekleyelim.









Form1 kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.Collections;

namespace WindowsFormsApplication11
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string AdSoyad = textBox1.Text; //Yerel degişkene bilgi textbox dan
alınıyor.

            Form2 YeniForm2 = new Form2(); //Projemize tasarım esnasında
oluşturduğumuz Form2 nin aynısında bir nesne (kopyasını) üretiyoruz. Bu YeniForm2,
Form2 nin tüm özelliklerini taşır. Çalışma esnasında artık bu yeni nesne
kullanılacak.
            YeniForm2.GelenBilgi = AdSoyad; //Form2 kodları arasında (From2.cs)
"GelenBilgi" public ifadesi kullanarak global tanımlamıştık. Dolayısıyla oradaki
tanımlama nedeniyle burada bu komutu görmektedir.
            YeniForm2.Show();               //Yeni Form2 açılıyor. YeniFrom2 yi
bilgiler degişkene yüklendikten sonra görüntülenmeli. yoksa değişkendeki bilgileri
göremeyiz.

            this.Hide(); //Bulunduğumuz form olan Form1 kapatıyoruz.

        }
    }
}
```

Form2 nin kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication11
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        public string GelenBilgi = null; //Bu degişkenin Form1 de görülebilmesi
için başında "public" ifadesinin olması gerekir.

        private void Form2_Load(object sender, EventArgs e)
        {
            label1.Text = GelenBilgi; //"GelenBilgi" değişkenine bilgi Form1 de
aktarılmıştı. public değişken (global) olduğu için oradan buraya gelişte bilgi
kaybolmamış oluyor.
        }
    }
}
```