**CONSOLE APP için DELEGATE KULLANIMI**

**Return kullanılarak ; Tek bir fonksiyon kullanımı**

namespace Delegate

{

class Program

{

**public delegate int Fonksiyon(int x, int y);**

static int Topla(int x,int y)

{

**return** x + y;

}

static int Carp(int a, int b)

{

**return** a \* b;

}

static void Main(string[] args)

{

int sonuc;

Fonksiyon **fnk** = new Fonksiyon(**Carp**);

sonuc=fnk(22,33);

Console.WriteLine("Sonuc:" + sonuc);

}

}

}

**Birden fazla fonksiyonun delegate’e atanıp kullanımı**

namespace Delegate

{

class Program

{

public delegate **void** Fonksiyon(int x, int y);

static void Topla(int x,int y)

{

Console.WriteLine(x + y);

}

static void Carp(int a, int b)

{

Console.WriteLine(a \* b);

}

static void Main(string[] args)

{

int sonuc;

**Fonksiyon fnk = new Fonksiyon(Carp);**

**fnk += Topla;**

fnk(22,33); //fnk.Invoke() fnk.BeginInvoke()

Console.WriteLine("Sonuc:");

}

}

}

**ÇIKTI:**

**726 (ÇARP)**

**55 (TOPLA)**

**FORM PROPERTIES YAZARAK PARAMETRE TRANSFERİ**

Form1 içinden Form2’yi çağırırken Form2’ye aktarılacak parametre,Form2 içinde tanımlanacak bir Property(get/set) ile Form2 içine çekilebilir. Bunun için öncelikli olarak Form2 içinde bir property tanımlanır ki property public erişimine sahip get ve set bloklarından oluşacaktır. Form1 içindeki bir textbox bilgisi Form2’nin oluşturulup çağrıldığı noktada , Form2 içinde tanımlı property çağırılarak Form2 içine aktarılır.

**Form2 property;**

public string DegisDokus

{

get

{

return deger;

}

set

{

value = deger;

}

}

**Form1 üzerinden çağrılma durumu;**

private void button1\_Click(object sender, EventArgs e)

{

Form2 frm2 = new Form2();

frm2.deger = textBox1.Text;

frm2.DegisDokus = textBox1.Text;

frm2.ShowDialog();

}

**APPLICATION.OPENFORMS ile PARAMETRE TRANSFERİ**

Application.OpenForms ile uygulamadaki açık formların bulunduğu koleksiyon kullanılarak (bu koleksiyon read-only koleksiyondur!!), bu koleksiyon içindeki çağırılan formun barındırdığı *Controls* koleksiyonundaki bir bileşene ulaşılabilir, değeri getirilebilir. *Application.OpenForms["FormName/indisi"].Controls.Find(“bilesenNAme”).FirstOrDefault()* metodu ile işlem gerçekleştirilir.

Uygulamada Form1 deki textbox içine bilgi yazılıp buton tıklandığında, Frm2 gösterilir. Frm2’nin Load event’i içine OpenForms kullanılarak Form1’deki textbox içeriği çekilebilir.

private void Form2\_Load(object sender, EventArgs e)

{

TextBox txt1=(TextBox)Application.OpenForms["Form1"].Controls.Find("textbox1", false).FirstOrDefault();

textBox1.Text = txt1.Text;

}

**FORM İÇİN DELEGATE KULLANIMI**

**Bir Formdan Diğerine veri aktarma.(delegate ve event kullanarak)**

Öncelikle Visual Studio da bir WindowsForms oluşturalım ve bu Form1 olsun.   
Formmumuza önce bir Listbox ekleyelim sonrada bir button ve textbox.   
daha sonra **SolutionExplorer>Add>WindowsForms** diyerek projemize yeni bir form ekleyelim. Bu formumuza ise textbox ve button ekleyelim. Şimdi yapacaklarımızı basit bir şekilde düşünelim. Form1 den button a basınca yeni form açılsın ve bizim textbox ın içine yazdığımızı açılan formun textbox ına yazsın. Sonra burda değiştirdiğimiz bu textbox ın içeriğini form2 de bulunan button ile form1 deki listbox’a yazalım. böylece form1 den form2 ye veri atmış ve form2 den de form1 e veri atmış oluyoruz.   
  
şimdi Form1 için yazmamız gereken kodlara geçelim.

**form1.cs**

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  //Burda form2 türünden yeni bir nesne oluşturuyoruz.  Form2 frm = new Form2();  //Burda Form2 altındaki textbox a erişip bu formdaki textbox ın değerini açacağımız formdakine yazıyoruz.  frm.texBox1.Text= textBox1.Text;  //burda olayımızı tetikliyoruz. ve olayımızn kullanacağı fonk gösteriyoruz.  frm.BenimEventim+= new Form2.BenimDelegem(VeriYaz);  //formumuzu gösteriyoruz. Burda frm.Show da kullanabilirdik . showdialog kullanarak  //sadece bir form açık ken buttona yeniden basılsada 2.bir form oluşturulmasını  //engelliyoruz  frm.ShowDialog();  }  //VeriYaz fonk tanımlanıyor. Fonksiyonumuz textbox tipinde değişken alıyor.  Void VeriYaz(TextBox txtBox)  {  // form2den gelen textbox değişkeninin değerini listbox a ekliyoruz.  Listbox1.Items.Add(txtBox.Text);  } |

**form2.cs**

|  |
| --- |
| // BenimDelegem isimli delegemizi tanımlıyoruz. Parametre olarak TextBox tipinde bir  //değişken alacağını söylüyoruz.  public delegate void BenimDelegem(TextBox txtbox);  // BenimDelegem tipinde BenimEventim isimli event oluşturuyoruz.  public event BenimDelegem BenimEventim;  private void button1\_Click(object sender, EventArgs e)  {  // buttona tıkladığımızda BenimEventim adlı olayımıza formumuzdaki textBox1 i atıyoruz.  this.BenimEventim(textBox1);  } |

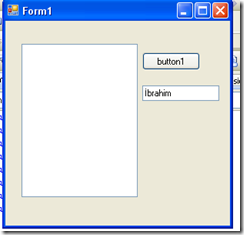
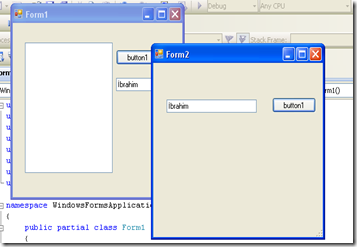
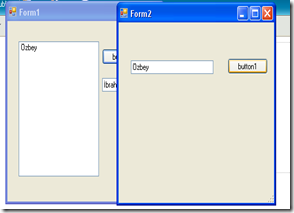
Şimdi Genel kurgumuzu yukarıda anlatmıştık fakat bu delegate ve event ne işe yarıyor birazda onlardan bahsedelim.   
burda form2 de önce bir delegate tanımlıyor ve bunun tipinde bir event oluştuyoruz.   
hemen aşağısında ise event olayında kullanacağımız form2 objesini olayımıza ekliyoruz.

**form1 de ise**

**frm.BenimEventim+= new Form2.BenimDelegem(VeriYaz);**

bu satırla form2 de tanımladığımız event e erişerek bu olayın *veriyaz* fonksiyonunu kullacağını söylüyor ve olayı tetikliyoruz.

Ayrıca **this.BenimEventim(textBox1);** kodu ile event imize attığımız textBox1 değerini Veriyaz Fonksiyonuna geçirmiş oluyoruz. ve bu değeri içerde kullanıyoruz.   
Aşağıda Projemizi ÇalıştırılMış Halini Görmektesiniz.

[](http://lh4.ggpht.com/_GSU6qHB3Rc4/SWfibnB2kUI/AAAAAAAAAHM/u1T7MQr95O8/s1600-h/image%5b13%5d.png)[](http://lh5.ggpht.com/_GSU6qHB3Rc4/SWfic-TejxI/AAAAAAAAAHU/pGUYJXafOhI/s1600-h/image%5b17%5d.png)[](http://lh4.ggpht.com/_GSU6qHB3Rc4/SWfid3D1_-I/AAAAAAAAAHc/t6ER5Q9C37c/s1600-h/image%5b20%5d.png)