

Nesneye Dayalı Programlama

Sakarya Üniversitesi
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği

Prof. Dr. Ümit KOCABIÇAK
Prof. Dr. Cemil Öz
Öğr. Gör. Nevzat TAŞBAŞI
Öğr. Gör. Sinan İLYAS

2017

13. HAFTA

TEMSİLCİ (DELEGATE)

Program yazarken bazen metotları değişkenlerde saklamak ya da başka bir metoda parametre olarak göndermek gerekebilir. Bu durumlarda temsilciler kullanılır.

```
delegate void Islem (int sayi);  
delegate double Topla (long sayi1, long sayi2);  
delegate string Oku();
```

TEMSİLCİ (DELEGATE)

Temsilciler, metotları saklayacak değişkenlerin şablonlarıdır.

Tanımlanan temsilciler, metotları saklayacak değişkenlerin veri tipi olarak kullanılır.

```
private delegate void Islem(int sayi);  
private Islem islemMetodu;
```

```
private delegate string Oku();  
private Oku okumaMetodu;
```

TEMSİLCİ (DELEGATE)

Temsilci tipli değişkenlerde metotlar saklanır.

```
private delegate void Komut();  
private Komut komut;
```

```
private void SelamVer()  
{  
    Console.WriteLine("Merhaba");  
}
```

...

```
komut = SelamVer; // Metot değişkende saklanıyor  
komut(); // Değişkende saklanan metot çalıştırılıyor
```

ÖRNEK-1

```
public delegate void Temsilci();

private Temsilci _temsilci;

public void SelamVer()
{
    MessageBox.Show("Merhaba.");
}

private void selamVerButton_Click(object sender, EventArgs e)
{
    _temsilci = SelamVer;
    _temsilci();
}
```

ÖRNEK-2

```
var yeniButton = new Button();  
  
yeniButton.Text = "Selam Ver";  
yeniButton.Left = 10;  
yeniButton.Top = 10;  
  
yeniButton.Click += SelamVer;  
  
Controls.Add(yeniButton);
```

ÖRNEK-3

```
private void SelamVer(object sender, EventArgs e)
{
    MessageBox.Show("Merhaba.");
}
```

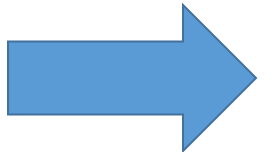
```
private void KendiniTanit(object sender, EventArgs e)
{
    MessageBox.Show("Ben Sinan.");
}
```

...

```
// Butona basılınca iki metot da çalışır.
tanisButton.Click += SelamVer;
tanisButton.Click += KendiniTanit;
```

ÖRNEK-4

```
private delegate void Komut();  
private Dictionary<string, Komut> komutlar;  
  
private void SelamVer()  
{  
    MessageBox.Show("Merhaba.");  
}  
  
private void KendiniTanit()  
{  
    MessageBox.Show("Ben Sinan.");  
}
```



ÖRNEK-4

```
komutlar = new Dictionary<string, Komut>();  
  
komutlar.Add("Selam Ver", SelamVer);  
komutlar.Add("Kendini Tanıt", KendiniTanıt);  
  
...  
  
_komutlar["Selam Ver"]();  
_komutlar["Kendini Tanıt"]();
```

ÖRNEK-5

```
private delegate void Temsilci(string mesaj);
```

```
private Temsilci _temsilci;
```

```
public void MesajGoster(string mesaj)
{
    MessageBox.Show(mesaj);
}
```

```
private void SelamVer()
{
    _temsilci = MesajGoster;
    _temsilci("Merhaba");
}
```

ÖRNEK-6

```
private delegate void Komut();

private void KomutCalistir(Komut komut)
{
    komut();
}

private void SelamVer()
{
    MessageBox.Show("Merhaba.");
}

private void selamVerButton_Click(object sender, EventArgs e)
{
    KomutCalistir(SelamVer);
}
```

OLAY (EVENT)

Olaylar bir nesneyi ilgilendiren herhangi bir durum olduğunda diğer nesneleri bilgilendirmek için kullanılır.

Olaylar en çok grafiksel kullanıcı arayüzlerinde karşımıza çıksa da kullanımı bununla sınırlı değildir.

ÖRNEK-1

```
class Dugme
{
    public event EventHandler TiklanmaOlayi;

    public void Tiklandi()
    {
        if (TiklanmaOlayi == null) return;
        TiklanmaOlayi(this, new EventArgs());
    }
}
```

...

```
var dugme = new Dugme();
dugme.TiklanmaOlayi += DugmeyeTiklandi;
dugme.Tiklandi();
```

ÖRNEK-2

```
private void tekKullanimlikButton_Click(object sender, EventArgs e)
{
    MessageBox.Show("Bu buton bir daha asla çalışmayacak.");

    tekKullanimlikButton.Click -= tekKullanimlikButton_Click;
}
```