

10. NDP Temel Kavram III: Çok Biçimlilik (Polymorphism)

Bir arayüzün ya da sınıfın kendisinden kalıtım alan sınıfları temsil edebilme yeteneği olarak tanımlayabileceğimiz “Çok Biçimlilik”, kalıtımın doğal sonucudur. Ancak kalıtım, geçersiz kılma ve geç bağlama yapı taşları olmadan da çok biçimlilik olamayacaktır.

Örnek 1: 2 ayrı sınıfın içerisinde barındırdığı alanları ekrana yazdıran program düşünelim.

```
class A
{
    private string adi = "A";

    public void EkranaYaz()
    {
        Console.WriteLine(this.adi);
    }
}
```

```
class B
{
    private string adi = "B";

    public void EkranaYaz()
    {
        Console.WriteLine(this.adi);
    }
}
```

```
class GenelSinif
{
    public void YazdirA()
    {
        A yeniA = new A();
        yeniA.EkranaYaz();
    }

    public void YazdirB()
    {
        yeniB = new B();
        yeniB.EkranaYaz();
    }
}
```

```

    }
}
class Program
{
    static void Main(string[] args)
    {
        GenelSinif g = new GenelSinif();
        g.YazdirA();
        g.YazdirB();
    }
}

```

Örneğimizde sadece 2 sınıf var. Eklenecek her yeni sınıf için birer ekrana yazdırma kodları ayrı ayrı metotlar ile oluşturulabilirdi. Halbuki EkranaYazdir() isimli bir metodumuz olsaydı ve gönderebileceğimiz parametreye göre A sınıfının ya da B sınıfının üyesini yazdırabilseydi NDP'nin gerçek hayat modellemesine daha yakın bir durum olurdu. Ancak bu noktada A ve B sınıflarının ortak bir yapısı olması gerekirdi.

Kodlarımızı aşağıdaki şekilde düzenleyelim.

```

abstract class BaseSinif
{
    public abstract void EkranaYaz();
}

class A : BaseSinif
{
    private string adi = "A";

    public override void EkranaYaz()
    {
        Console.WriteLine(this.adi);
    }
}

class B : BaseSinif
{
    private string adi = "B";

    public override void EkranaYaz()
    {
        Console.WriteLine(this.adi);
    }
}

```

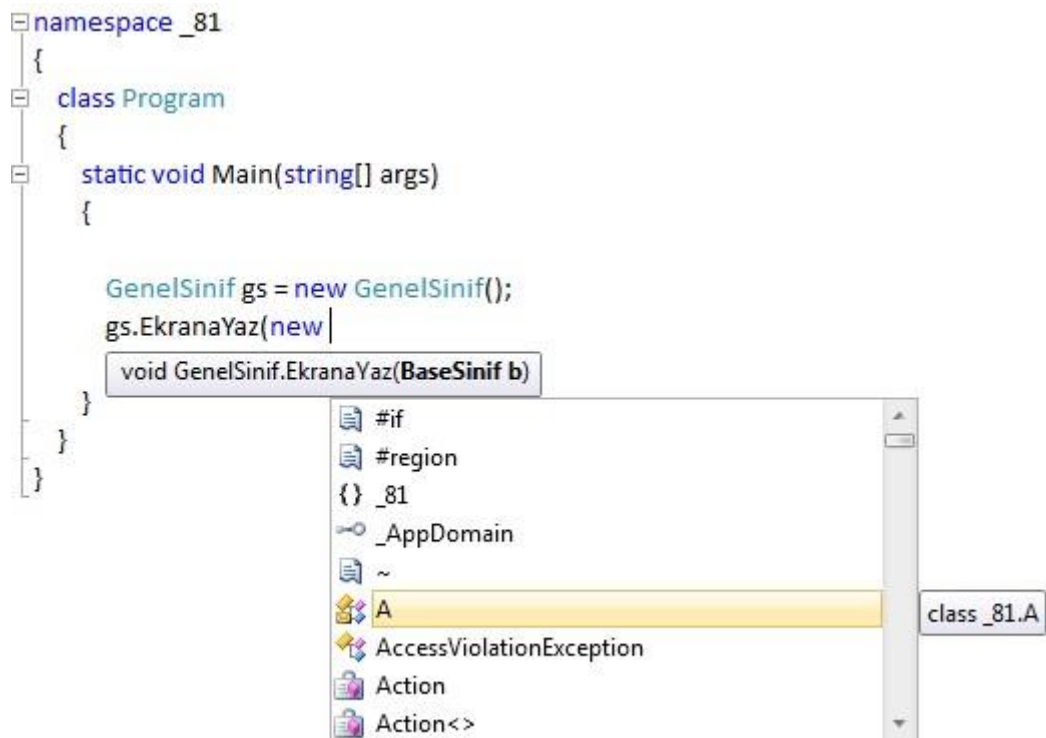
```

    }
}

class GenelSinif
{
    public void EkranaYaz(BaseSinif b) // temel sınıf türünden
    parametre

    {
        b.EkranaYaz();
    }
}

```



GenelSinif'tan oluşan gs isimli nesnenin EkranaYaz() isimli metodu parametre olarak BaseSinif() türünden bir yapı beklemektedir. Böylelikle ister A sınıfını ister B sınıfını dilersekde yine BaseSinif isimli sınıftan kalıtım alan herhangi bir sınıfı parametre olarak kullanabiliriz.

```

class Program
{
    static void Main(string[] args)
    {
        GenelSinif gs = new GenelSinif();
        gs.EkranaYaz(new A());
        gs.EkranaYaz(new B());
    }
}

```



Örnek 2: Hayvan seslerini ekrana yazdıran program.

```

abstract class HayvanBase
{
    private string adi;
    public string Adi
    {
        get
        {
            return adi;
        }
    }
    public HayvanBase(string adi)
    {
        this.adi = adi;
    }
    public abstract string SesVer();
}

class Kedi : HayvanBase
{
    public Kedi(string adi)
        : base(adi)
    {

```

```

    }

    public override string SesVer()
    {
        return "Miyaw";
    }
}

class Kopek : HayvanBase
{
    public Kopek(string adi)
        : base(adi)
    {
    }

    public override string SesVer()
    {
        return "Hav";
    }
}

class Program
{
    static void Main(string[] args)
    {
        HayvanBase[] bakimEvi = { new Kopek("Karabaş"), new
Kedi("Pamuk") };

        foreach (HayvanBase h in bakimEvi)
            Console.WriteLine(h.SesVer());
    }
}

```

Örnek 3: Hoca ve sınıfları ekrana yazdıran program.

```
abstract class İnsanBase
{
    protected string adi;
    protected string soyadi;
    public İnsanBase(string adi, string soyadi)
    {
        this.adi = adi;
        this.soyadi = soyadi;
    }

    public virtual string Etiket()
    {
        return this.adi + " " + this.soyadi;
    }
}

class Sinif : İnsanBase
{
    private string m_ogrNo;
    public string OgrNo
    {
        set { m_ogrNo = value; }
    }

    public Sinif(string ogrNo, string adi, string soyadi) :
base(adi, soyadi)
    {
        this.m_ogrNo = ogrNo;
    }
    public override string Etiket()
    {
        return this.m_ogrNo + " " + this.adi + " " + this.soyadi;
    }
}

class Hoca : İnsanBase
{
    public Hoca(string ad, string soyAdi)
        : base(ad, soyAdi)
    {
    }
}

class Program
{
    static void Main(string[] args)
```

```

{
    Hoca c1 = new Hoca("Özgür", "Çiftçi");
    Hoca c2 = new Hoca("Hüseyin", "Eski");
    Hoca c3 = new Hoca("Kayhan", "Ayar");

    Sinif gunduzA = new Sinif("1A", "1. Öğretim", "A");
    Sinif gunduzB = new Sinif("1B", "1. Öğretim", "B");
    Sinif geceA = new Sinif("2A", "2. Öğretim", "A");
    Sinif geceB = new Sinif("2B", "2. Öğretim", "B");
    Sinif karma = new Sinif("Karma", "Karma", "A");

    İnsanBase[] m = { c1, c2, c3, gunduzA, gunduzB, geceA,
geceB, karma };

    foreach (İnsanBase yeni in m)
        EkranaYazdir(yeni);
}
static void EkranaYazdir(İnsanBase i)
{
    Console.WriteLine(i.Etiket());
}
}

```