

## NYP 5. ÖDEV

### 1- Delegate ve Özel Delege Tanımlama:

Bir delegate tanımlayarak, iki sayıyı çıkaran bir metot yazın. Aynı işlevi yerine getiren özel bir delege tanımlayın. Bu delegeyi kullanarak iki farklı metodu temsil eden bir dizi oluşturun: birincisi sayıları çıkaran, ikincisi sayıları toplayan.

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

namespace NypOdev5
{
    class Program
    {
        // Delegate tanımlama
        public delegate int Islem(int x, int y);

        // Delegate'i kullanan metot
        public static int Fark(int x, int y)
        {
            return x-y;
        }

        // Özel delegate tanımlama
        public delegate int Islem2(int x, int y);

        // Özel delegate'i kullanan metot
        public static int Toplam(int x, int y)
        {
            return x + y;
        }

        static void Main(string[] args)
        {
            // Delegate dizi oluşturma
            Islem2[] islemler = new Islem2[2];

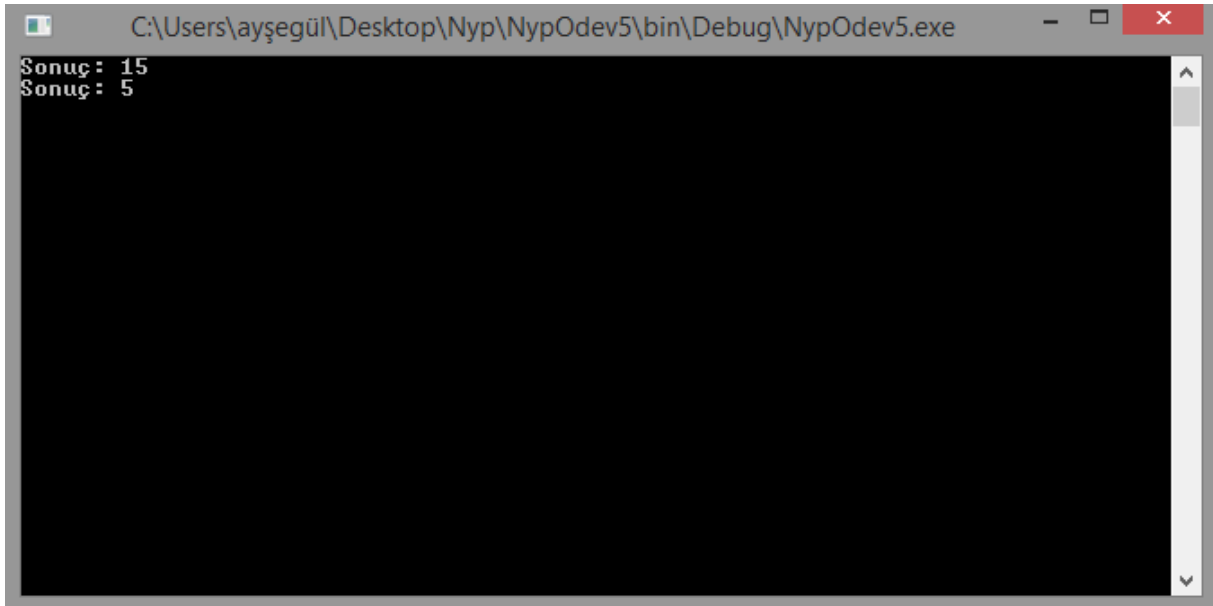
            // Dizinin elemanlarını delegate'lerle doldurma
            islemler[0] = Toplam;
            islemler[1] = Fark;

            // Dizideki her delegate'i tetikleme
            foreach (Islem2 islem in islemler)
            {
                int sonuc = islem(10, 5); islem(10, 5);
            }
        }
    }
}
```

```

        Console.WriteLine("Sonuç: " + sonuc);
    }
    Console.ReadLine();
}
}
}

```



### Events ve Event Argümanları:

Bir event argümanı kullanarak, bir kitabın özelliklerini içeren bir BookEventArgs sınıfı oluşturun. Bir kütüphane sınıfı oluşturun ve bu sınıfın kitap eklenmesini temsil eden bir event tanımlayın. Event abonesi olarak bir başka sınıf oluşturun. Bu sınıf, eklenen kitabın adını, yazarını ve yayın yılını ekrana yazdırsın. Event yayıcısı sınıfı içinde, kitap eklenmesi durumunda event'i tetikleyen bir metot ekleyin.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace NypOdev5
{
    class Program
    {
        public class BookEventArgs : EventArgs

```

```

{
    public string Ad { get; set; }
    public string Yazar { get; set; }
    public int YayinYili { get; set; }

    public BookEventArgs(string ad, string yazar, int yayinYili)
    {
        Ad = ad;
        Yazar = yazar;
        YayinYili = yayinYili;
    }
}

public class Kutuphane
{
    public event EventHandler<BookEventArgs> KitapEklendi;

    public void KitapEkle(string ad, string yazar, int yayinYili)
    {
        // Kitap ekleme işlemleri

        // Event'i tetikleme
        KitapEklendi?.Invoke(this, new BookEventArgs(ad, yazar, yayinYili));
    }
}

public class KitapAbonesi
{
    public void AboneKitapEklendi(object sender, BookEventArgs e)
    {
        Console.WriteLine("Kitap Eklendi:");
        Console.WriteLine("Ad: " + e.Ad);
        Console.WriteLine("Yazar: " + e.Yazar);
        Console.WriteLine("Yayın Yılı: " + e.YayinYili);
    }
}

static void Main(string[] args)
{
    Kutuphane kutuphane = new Kutuphane();
    KitapAbonesi abone = new KitapAbonesi();

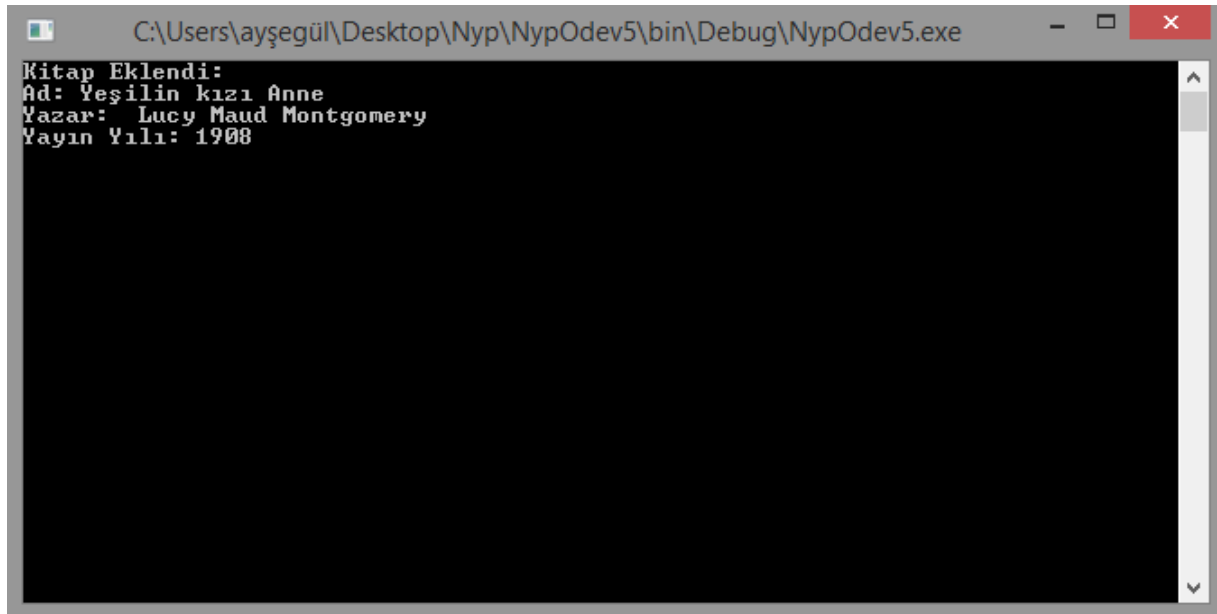
    // Event aboneliği
    kutuphane.KitapEklendi += abone.AboneKitapEklendi;

    // Kitap ekleme
    kutuphane.KitapEkle("Yeşilin kızı Anne ", " Lucy Maud Montgomery", 1908);

    Console.ReadLine();
}
}

```

}



#### Delegate ve Events Uygulaması:

Bir string listesini işleyen ve her elemanın uzunluğunu ekrana yazdıran bir StringProcessor sınıfı oluşturun. Bu sınıf, işlem sırasında her elemanı bir delegate vasıtasıyla modifiye edebilmelidir. İki farklı delegate tanımlayın: birincisi string elemanları ters çeviren, ikincisi string elemanlarını büyük harfe çeviren birer metodu temsil etmelidir. StringProcessor sınıfını kullanarak, bir liste üzerinde bu iki farklı delegate'i kullanarak işlem yapın ve sonuçları ekrana yazdırın.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace NypOdev5
{
    class Program
    {
        public class StringProcessor
        {
            public delegate string StringIslem(string str);

            public void Isle(List<string> kelimeler, StringIslem islem)
            {
                foreach (string kelime in kelimeler)
                {
```

```

        Console.WriteLine(islem(kelime) + "(" + kelime.Length + ")");
    }
}

public string TersCevir(string str)
{
    char[] dizi = str.ToCharArray();
    Array.Reverse(dizi);
    return new string(dizi);
}

public string BuyukHarf(string str)
{
    return str.ToUpper();
}

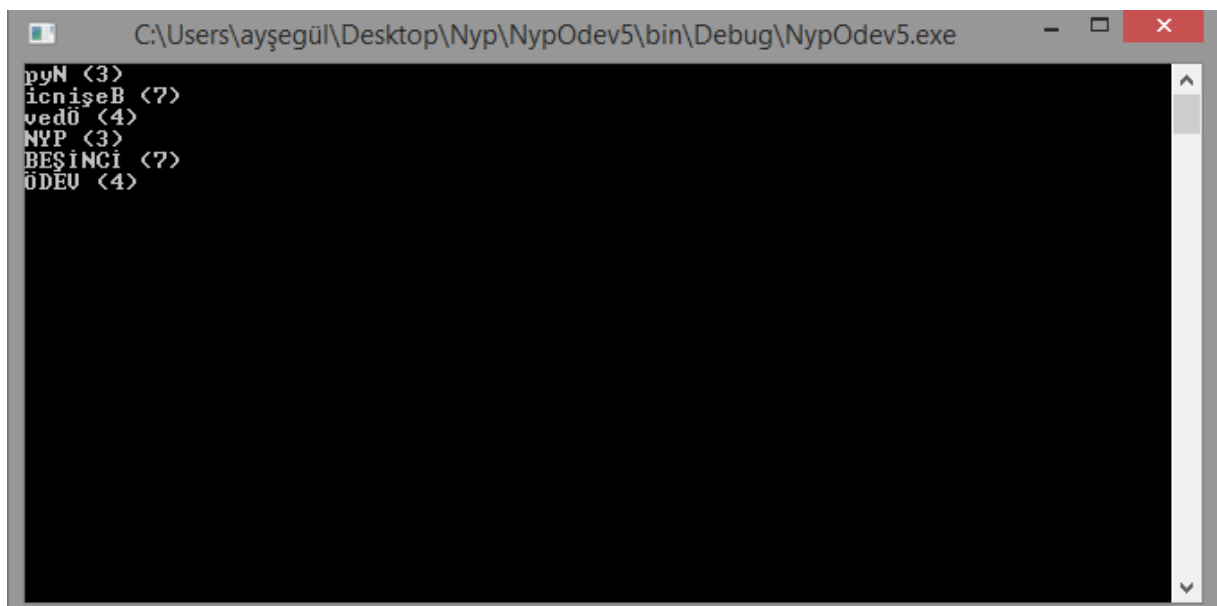
static void Main(string[] args)
{
    StringProcessor processor = new StringProcessor();
    List<string> kelimeler = new List<string>() { "Nyp", "Beşinci", "Ödev" };

    // Ters çevirme işlemi
    processor.Isle(kelimeler, processor.TersCevir);

    // Büyük harf çevirme işlemi
    processor.Isle(kelimeler, processor.BuyukHarf);

    Console.ReadLine();
}
}
}

```



```

C:\Users\ayşegül\Desktop\Nyp\NypOdev5\bin\Debug\NypOdev5.exe
pyN <3>
icnişeb <7>
vedö <4>
NYP <3>
BEŞİNCİ <7>
ÖDEV <4>

```

2- Bir hesap makinesi sınıfı oluşturun. Bu sınıf, iki sayıyı toplama, çıkarma, çarpma ve bölme işlemlerini gerçekleştirebilmelidir. Bu işlemleri temsil eden dört farklı delegate tanımlayın ve bu delegate'leri kullanarak hesaplamalar gerçekleştirin. Kullanıcıdan iki sayı ve bir işlem tipi alın ve ilgili delegate'i kullanarak sonucu ekrana yazdırın.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace NypOdev5
{
    class Program
    {
        // Delegate tanımları
        public delegate double HesaplamaDelegate(double sayi1, double sayi2);

        // Hesap makinesi sınıfı
        public class HesapMakinesi
        {
            // Toplama işlemi delegate'i
            public HesaplamaDelegate ToplaDelegate = (sayi1, sayi2) => sayi1 + sayi2;

            // Çıkarma işlemi delegate'i
            public HesaplamaDelegate CikarDelegate = (sayi1, sayi2) => sayi1 - sayi2;

            // Çarpma işlemi delegate'i
            public HesaplamaDelegate CarpDelegate = (sayi1, sayi2) => sayi1 * sayi2;

            // Bölme işlemi delegate'i
            public HesaplamaDelegate BolDelegate = (sayi1, sayi2) => sayi1 / sayi2;
        }

        static void Main(string[] args)
        {
            HesapMakinesi hesapMakinesi = new HesapMakinesi();

            Console.WriteLine("Birinci sayıyı girin: ");
            double sayi1 = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("İkinci sayıyı girin: ");
            double sayi2 = Convert.ToDouble(Console.ReadLine());

            Console.WriteLine("İşlemi seçin (+, -, *, /): ");
            char islemTipi = Console.ReadKey().KeyChar;
            Console.WriteLine(); // Yeni satıra geç

            HesaplamaDelegate hesaplamaDelegate = null;
```

```

// Kullanıcının seçtiği işleme göre ilgili delegate'i seçme
switch (islemTipi)
{
    case '+':
        hesaplamaDelegate = hesapMakinesi.ToplaDelegate;
        break;
    case '-':
        hesaplamaDelegate = hesapMakinesi.CikarDelegate;
        break;
    case '*':
        hesaplamaDelegate = hesapMakinesi.CarpDelegate;
        break;
    case '/':
        hesaplamaDelegate = hesapMakinesi.BolDelegate;
        break;
    default:
        Console.WriteLine("Hata: Geçersiz işlem tipi.");
        return;
}

// Hesaplama yapma ve sonucu ekrana yazdırma
double sonuc = hesaplamaDelegate(sayi1, sayi2);
Console.WriteLine($"Sonuç: {sonuc}");
Console.ReadLine();
}
}
}

```

```

C:\Users\ayşegül\Desktop\Nyp\NypOdev5\bin\Debug\NypOdev5.exe
Birinci sayıyı girin: 7
İkinci sayıyı girin: 6
İşlemi seçin (<+, -, *, />): *
Sonuç: 42

```

### 3- Exception Handling (Hata Yönetimi):

Bir hesap makinesi uygulaması oluşturun. Kullanıcıdan iki sayı ve bir işlem türü (toplama, çıkarma, çarpma, bölme) alın. Kullanıcı eğer bir sayıyı sıfıra bölmeye çalışırsa, bir

DivideByZeroException fırlatın ve hatayı yakalayıarak kullanıcıya uygun bir mesaj gösterin. Kullanıcının geçersiz bir işlem türü girmesi durumunda ise, InvalidOperationException fırlatın ve hatayı yakalayıarak kullanıcıya uygun bir mesaj gösterin.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace NypOdev5
{
    class Program
    {
        public class HesapMakinesi
        {
            public int Hesapla(int x, int y, string islemTipi)
            {
                try
                {
                    switch (islemTipi)
                    {
                        case "+":
                            return x + y;
                        case "-":
                            return x - y;
                        case "*":
                            return x * y;
                        case "/":
                            if (y == 0)
                            {
                                throw new DivideByZeroException("Sıfıra bölme işlemi yapılamaz!");
                            }
                            return x / y;
                        default:
                            throw new InvalidOperationException("Geçersiz işlem tipi!");
                    }
                }
                catch (DivideByZeroException ex)
                {
                    Console.WriteLine(ex.Message);
                }
                catch (InvalidOperationException ex)
                {
                    Console.WriteLine(ex.Message);
                }
                catch (Exception ex)
                {
                    Console.WriteLine("Beklenmedik bir hata oluştu!");
                }
            }
        }
    }
}
```



```

        return 0;
    }
}
static void Main(string[] args)
{
    HesapMakinesi hesapMakinesi = new HesapMakinesi();

    Console.WriteLine("Birinci sayıyı girin:");
    int x = int.Parse(Console.ReadLine());

    Console.WriteLine("İkinci sayıyı girin:");
    int y = int.Parse(Console.ReadLine());

    Console.WriteLine("İşlemi seçin (+, -, *, /): ");
    string islemTipi = Console.ReadLine();

    int sonuc = hesapMakinesi.Hesapla(x, y, islemTipi);

    Console.WriteLine("Sonuç: " + sonuc);
    Console.ReadLine();
}
}
}

```

```

C:\Users\ayşegül\Desktop\Nyp\NypOdev5\bin\Debug\NypOdev5.exe
Birinci sayıyı girin:
25
İkinci sayıyı girin:
5
İşlemi seçin (<+, -, *, />):
/
Sonuç: 5

```

### Attributes (Öznitelikler):

Bir öğrenci sınıfı oluşturun. Öğrenci sınıfına IsApproved adında bir öznitelik (attribute) ekleyin. Bu öznitelik, öğrencinin başarılı olup olmadığını belirleyen bir boolean değeri

içermelidir. Öğrenci sınıfını kullanarak birkaç öğrenci oluşturun ve IsApproved öz niteliklerini kullanarak başarılı öğrencileri filtreleyin.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Threading.Tasks;
namespace NypOdev5
{
    class Program
    {
        [AttributeUsage(AttributeTargets.Class)]
        public class IsApprovedAttribute : Attribute
        {
            public bool IsApproved { get; set; }

            public IsApprovedAttribute(bool isApproved)
            {
                IsApproved = isApproved;
            }
        }
        [IsApprovedAttribute(true)]
        public class Ogrenci
        {
            public string Ad { get; set; }
            public string Soyad { get; set; }
            public int VizeNotu { get; set; }
            public int FinalNotu { get; set; }

            public Ogrenci(string ad, string soyad, int vizeNotu, int finalNotu)
            {
                Ad = ad;
                Soyad = soyad;
                VizeNotu = vizeNotu;
                FinalNotu = finalNotu;
            }

            public bool IsApproved
            {
                get
                {
                    return (VizeNotu + FinalNotu) / 2 >= 50;
                }
            }

            public override string ToString()
            {

```

```

        return $"{Ad} {Soyad} - Vize: {VizeNotu} - Final: {FinalNotu} - Başarılı:
{IsApproved}";
    }
}
static void Main(string[] args)
{
    // Öğrenci nesneleri oluşturma
    Ogrenci ogr1 = new Ogrenci("Ayşegül", "Saygı", 60, 70);
    Ogrenci ogr2 = new Ogrenci("Ahmet", "Saygı", 40, 40);

    // IsApproved öz niteliğini kullanarak başarılı öğrencileri filtreleme
    List<Ogrenci> basariliOgrenciler = new List<Ogrenci>();
    foreach (var ogr in new Ogrenci[] { ogr1, ogr2})
    {
        if (ogr.GetType().GetCustomAttribute<IsApprovedAttribute>()?.IsApproved ??
false)
        {
            basariliOgrenciler.Add(ogr);
        }
    }

    // Başarılı öğrencileri yazdırma
    foreach (var ogr in basariliOgrenciler)
    {
        Console.WriteLine(ogr);
    }
    Console.ReadLine();
}
}
}

```

```

C:\Users\ayşegül\Desktop\Nyp\NypOdev5\bin\Debug\NypOdev5.exe
Ayşegül Saygı - Vize: 60 - Final: 70 - Başarılı: True
Ahmet Saygı - Vize: 40 - Final: 40 - Başarılı: False

```

## Reflection (Yansıma):

Bir sınıf oluşturun ve bu sınıfa birkaç özellik (property) ekleyin. Reflection kullanarak, bu sınıfın adını, özelliklerini ve bu özelliklere ait değerleri ekrana yazdıran bir metod yazın. Oluşturduğunuz sınıfı kullanarak bir nesne oluşturun ve bu nesnenin özelliklerini yansıma kullanarak ekrana yazdırın.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Threading.Tasks;
namespace NypOdev5
{
    class Program
    {
        class Ogrenci
        {
            // Özellikler (Properties)
            public string Ad { get; set; }
            public string Soyad { get; set; }
            public double SinavNotu { get; set; }
            public Ogrenci(string ad,string soyad,double sinav)
            {
                Ad = ad;
                Soyad = soyad;
                SinavNotu = sinav;
            }

            // Yansıma kullanarak bilgileri ekrana yazdıran metod
            public void OzellikleriYazdir()
            {
                Type type = this.GetType();
                PropertyInfo[] properties = type.GetProperties();

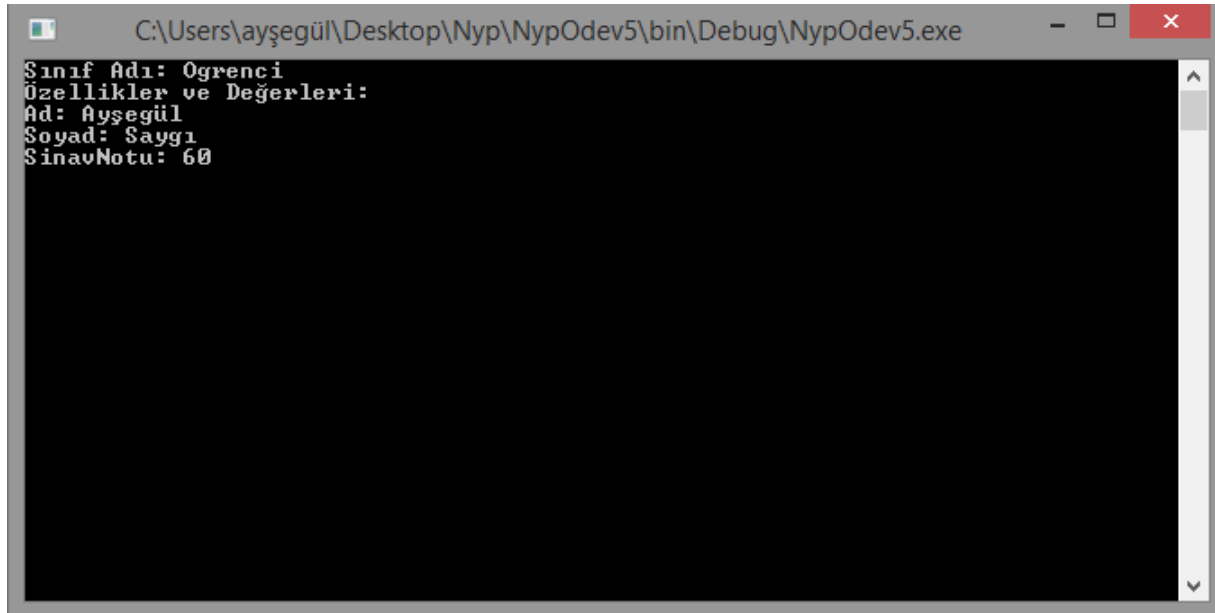
                Console.WriteLine("Sınıf Adı: " + type.Name);
                Console.WriteLine("Özellikler ve Değerleri:");

                foreach (var property in properties)
                {
                    object value = property.GetValue(this);
                    Console.WriteLine($"{property.Name}: {value}");
                }
            }
        }

        static void Main(string[] args)
        {

```

```
Ogrenci ogrenci = new Ogrenci("Ayşegül", "Saygı", 60);
ogrenci.OzellikleriYazdir();
Console.ReadLine();
    }
}
}
```



```
C:\Users\ayşegül\Desktop\Nyp\NypOdev5\bin\Debug\NypOdev5.exe
Sınıf Adı: Ogrenci
Özellikler ve Değerleri:
Ad: Ayşegül
Soyad: Saygı
SınavNotu: 60
```