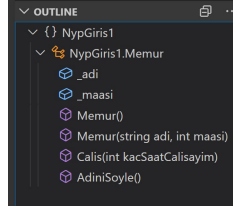


NYP Prensipleri (1-Kapsülleme)

15 Aralık 2019 Pazar 23:39

- NYP Prensipleri
- Değişken ve property(özelliği)ler arasındaki fark
- Kapsülleme işlemi
- Bazı String işlemleri

```
Memur.cs > {} NypGiris1 > NypGiris1.Memur > Memur(string adi, int maasi)
1 using System;
2 namespace NypGiris1
3 {
4     class Memur
5     {
6         public string _adi;
7         public int _maasi;
8
9         public Memur()
10        {
11            Console.WriteLine("Yuppiiii... Memur oldum!");
12        }
13
14        public Memur(string adi, int maasi)
15        {
16            _adi = adi;
17            _maasi = maasi;
18            Console.WriteLine("Yuppiiii... Memur oldum!");
19            this.AdiniSoyle();
20        }
21
22        public void Calis(int kacSaatCalisayim)
23        {
24            //Calismakla ilgili kodlar yazilacak
25            Console.WriteLine("{0} saat çalıştım ve çok yoruldum", kacSaatCalisayim);
26        }
27
28        public void AdiniSoyle()
29        {
30            Console.WriteLine("Ben memur {0}, ayda {1}TL kazanıyorum", _adi, _maasi);
31        }
32    }
33 }
```

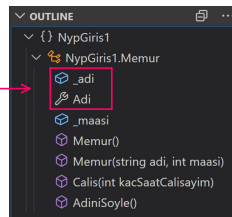


```
Memur.cs > {} NypGiris1 > NypGiris1.Memur
1 using System;
2 namespace NypGiris1
3 {
4     class Memur
5     {
6         prop
7         pub | prop
8         | propfull
9         | propg
10        {
11            InvalidProgramException
12            PlatformNotSupportedException
13        }
14
15        public Memur(string adi, int maasi)
16        {
17            _adi = adi;
18            _maasi = maasi;
19            Console.WriteLine("Yuppiiii... Memur oldum!");
20        }
21    }
22 }
```

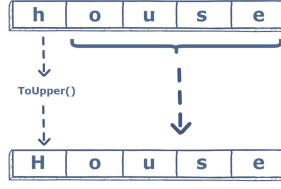
```
Property and backing field (C#)
private int myVar;
public int MyProperty
{
    get { return myVar; }
    set { myVar = value; }
}
```

```
Memur.cs > {} NypGiris1 > NypGiris1.Memur > Calis(int kacSaatCalisayim)
1 using System;
2 namespace NypGiris1
3 {
4     class Memur
5     {
6         private string _adi;
7         public string Adi
8         {
9             get { return _adi; }
10            set { _adi = value; }
11        }
12
13        public int _maasi;
14
15        public Memur()
16        {
17            Console.WriteLine("Yuppiiii... Memur oldum!");
18        }
19
20        public Memur(string adi, int maasi)
21        {
22            _adi = adi;
23            _maasi = maasi;
24            Console.WriteLine("Yuppiiii... Memur oldum!");
25            this.AdiniSoyle();
26        }
27
28        public void Calis(int kacSaatCalisayim)
29        {
30            //Calismakla ilgili kodlar yazilacak
31            Console.WriteLine("{0} saat çalıştım ve çok yoruldum", kacSaatCalisayim);
32        }
33
34        public void AdiniSoyle()
35        {
36            Console.WriteLine("Ben memur {0}, ayda {1}TL kazanıyorum", _adi, _maasi);
37        }
38    }
39 }
```

Handwritten notes:
- "public string Adi" is circled in blue and labeled "Değişken".
- "private string _adi" is circled in green and labeled "Property".
- "get { return _adi; }" and "set { _adi = value; }" are circled in red and labeled "Property işlemleri".
- A pink arrow points from the "Adi" property to the "get/set" methods.
- To the right, there are handwritten examples: "x = 5;" and "x = 6;" with a pink arrow pointing to the "Adi" property in the Outline window.



```
private string _adi;
public string Adi
{
    get { return _adi; }
    set
    {
        if (value.Length >= 2)
            _adi = Char.ToUpper(value[0]) + value.Substring(1).ToLower();
        else
            _adi = "";
    }
}
```



```
4 references
private string _adi;
1 reference
public string Adi //Adi <-> value
{
    get { return _adi; } //korunmalı değişkeni (_adi) elde etmeye çalıştığımız -okumaya- zaman
    // set { _adi = value; } //korunmalı değişkeni (_adi) ayarlamaya çalıştığımız -yazmaya- zaman
    set
    {
        if (value.Length <= 1) _adi = "Hatallı İsim"; //Tek karakterlik isim olamayacağı için
        // else _adi = value;
        else _adi = char.ToUpper(value[0]) + value.Substring(1).ToLower();
    }
}
```

```
class Program
{
    static void Main(string[] args) //method
    {
        Memur memur1 = new Memur("AHMET", 2500);
        memur1.Calis(8);

        Console.WriteLine("-----");

        Memur memur2 = new Memur("mEHMet", 6000);
        memur2.Calis(1);
    }
}
```

```
Yuppiiii... Memur oldum!
Ben memur Ahmet, ayda 2500TL kazanıyorum
8 saat çalıştım ve çok yoruldum
-----
Yuppiiii... Memur oldum!
Ben memur Mehmet, ayda 6000TL kazanıyorum
1 saat çalıştım ve çok yoruldum
```

Sp6+

Uygulama: AdıSoyadı -> "AHMET ÇOKÇALIŞIR" -> "Ahmet ÇOKÇALIŞIR"

Uygulama: DoğumYılı -> Günümüzden 18 yıl öncesi değilse varsayılan olarak günümüzden 18 yıl öncesi,

iki karakter girilirse (ör: 80 -> 1980) 1900 ile birleştirilmeli

Uygulama: Sicil Notu -> 0..100 arasında değilse 0 kabul edecek -30 -> 0

Uygulama: Memurun maaşının %20 kesintili halini veren "NetMaas" isimli ve sadece okunabilir olan (ayarlanamayan) property'i yazınız.