Object Oriented Programming

Class Members

Field

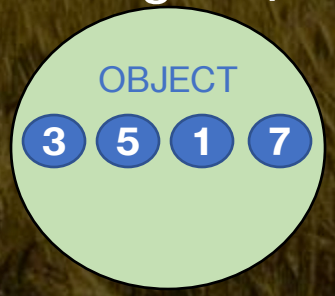
Field'lar türüne özgü varsayılan değer alırlar.

```
class MyClass
{
    int a;
}
```

STACK	
a	0

Eğer bir değişken class içerisinde field olarak tanımlanıyorsa default değeri verilir. Yok eğer class'ta değil metot vs. içerisinde tanımlanıyorsa defaul değer verilmez!

- Nesne içerisinde veri depoladığımız/tuttuğumuz alanlardır.



- Class içerisindeki değişkenlerdir.
- Herhangi bir türden olabilir.

Property

Property'nin işlevsel açıdan metottan farkı yoktur, lakin davranışsal olarak nesne üzerinde bir değer okuma ve değer atama işlemlerinde kullanılır.

Peki nesne içerisinde özellik/property sağlaması ne demektir?

- Nesne içerisinde özellik/property sağlar.
- Property esasında özünde bir metottur. Yani programatik/algoritmik kodlarımızı inşa ettiğimiz bir metot.
- Lakin fiziksel olarak metottan farkı parametre almamakta ve içerisinde get ve set olmak üzere iki adet blok almaktadır.
- Keza bu bloklar compile neticesinde get ve set isimli metotlar olarak karşımıza çıkmaktadır.

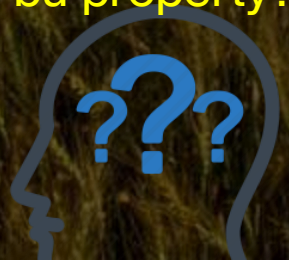
```
public int X()
{
    return 0;
}
```

```
public int X
{
    get
    {
        return 0;
    }
    set
    {
        //Atanan veri buradan yakalanır.
    }
}
```

Property'nin değeri çağrıldığında get bloğu tetiklenir ve değeri return eder.

Property'e bir değer atandığında o değeri set bloğu karşılar

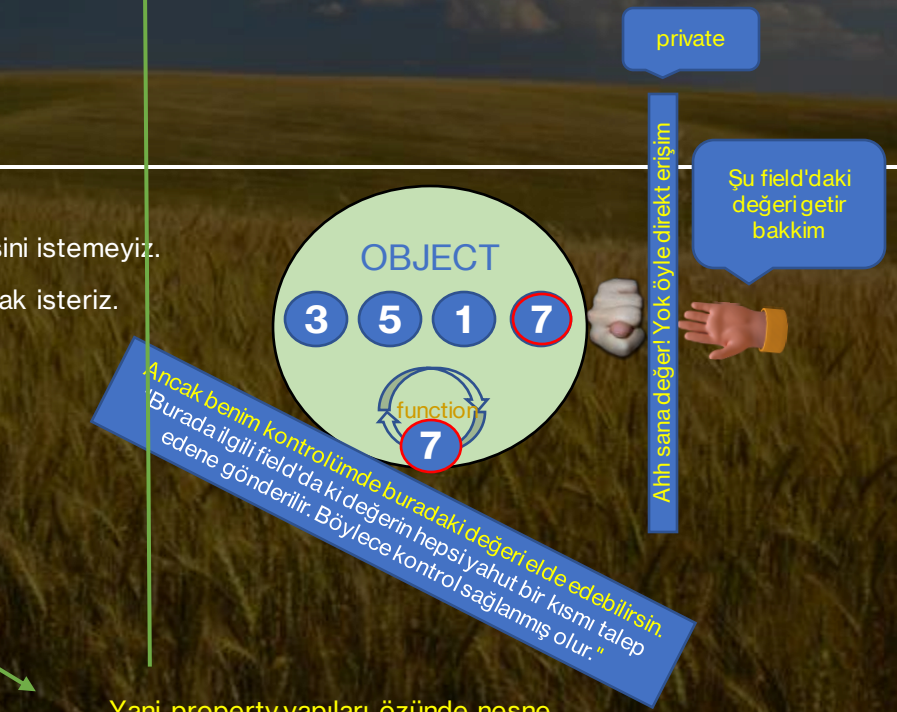
Peki ne işe yaramakta bu property?



İşte... Biz property'lerin bu
işlevine Encapsulation(Kapsülleme/Sarmalama) diyeceğiz...

Property

- Biz yazılımcılar nesnelerimiz içerisindeki field'lara direkt erişilmesini istemeyiz.
- Dolayısıyla field'lar da ki verileri kontrollü bir şekilde dışarıya açmak isteriz.
- İşte böyle bir durumda metotları kullanabiliriz.



Böyle bir durumda C# programlama dilinde metot yerine property yapıları geliştirilmiştir.

Yani property yapıları özünde nesne içerisindeki bir field'ın dışarıya kontrollü açılmasını ve kontrollü bir şekilde dışarıdan değer almasını sağlayan yapılardır.

İşte bu şekilde field'larda ki verilerin erişim kontrolünü yapmamız için geliştirilmiş olan yapılara Property denmektedir.

Encapsulation(Kapsülleme/Sarmalama)

- Encapsulation, bir nesne içerisindeki dataların(field'lardaki verilerin) dışarıya kontrollü bir şekilde açılması ve kontrollü bir şekilde veri almasıdır.

Property İmzaları

- Property yapısı oluşturabilmenin yapısal olarak birkaç farklı yolu/farklı imzası vardır.
  - Full Property
  - Prop
  - Auto Property Initializers
  - Ref Readonly Returns
  - Computed(Hesaplanmış) Properties
  - Expression-Bodied Property
    - Read Only Property
  - Init-Only Properties ve Init Accessor

Full Property

- En sade property yapılanmasıdır.
- İçerisinde get ve set blokları tanımlanmalıdır.

```
[erişim belirleyicisi] [geri dönüş değeri] [property adı]
{
        get {       }   → Property'den veri istendiğinde tetiklenir.

        set {       }   → Property'e veri gönderildiğinde tetiklenir.
                          Gönderilen veriyi value keywordüyle
                          yakalar.
}
```

Full propertylerde set bloğu tanımlanmazsa sadece okunabilir(read only) bilakis get bloğu tanımlanmazsa sadece yazılabilir(write only) olacaktır.

Hayır! Field'da ki değere müdahale olsun olmasın direkt erişim yapılmasını istemiyoruz. Bu alışkanlığımız olsun.
Haliyle böyle bir durumda yine property kullanacağız. Sadece get ve set blokları aşağıdaki gibi tanımlanması yeterli olacaktır.

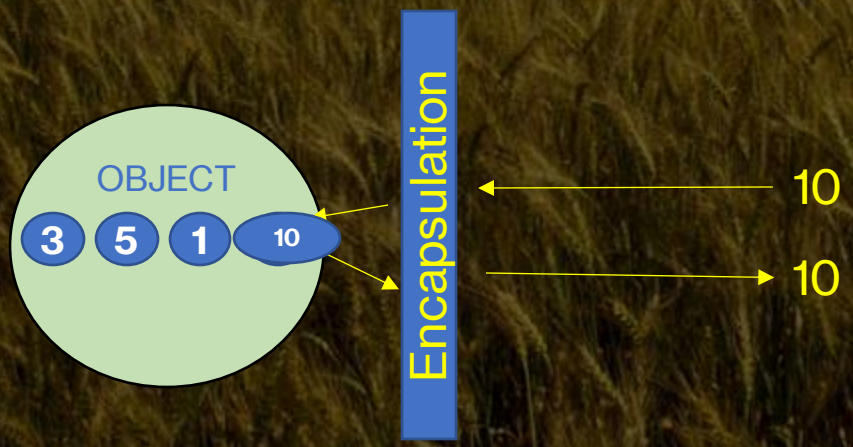
Prop property'ler compile edildiklerinde arkaplanda kendi field'larını oluştururlar. Dolayısıyla bir field tanımlamaya gerek yoktur!

PROP

```
[erişim belirleyicisi] [geri dönüş değeri] [property adı]{ get; set; }
```

Prop imzalarda ilgili property read only olabilir lakin write only olamaz!

- Bir property her ne kadar encapsulation yapsada temsil ettiği field'da ki dataya hiç müdahale etmeden erişilmesini ve veri atanmasını sağlıyorsa böyle bir durumda kullanılan property imzasıdır.



Hocam! Böyle bir durumda direkt field'a erişim sağlasak!

Auto property initializers özelliği sayesinde read only olan prop'lara hızlıca değer atanabilmektedir.

Auto Property Initializers (C# 6.0)

- Bir property'nin ilk değerini nesne ayağa kaldırılır kaldırılmaz aşağıdaki gibi verebiliriz.

```
class InsanEntity
{
    public string Adi { get; set; } = "Gençay";
    public string Soyadi { get; set; } = "Yıldız";
    public int Yasi { get; set; } = 23;
}
```

Ref Readonly Returns

- ref readonly returns, bir sınıf(class) içerisindeki field'ı referansıyla döndürmemizi sağlayan ve biryandan da bu değişkenin değerini read only yapan özelliktir.

Computed(Hesaplanmış) Properties

- İçerisinde türetilmiş bir bağıntı taşıyan property'lerdir.

Expression-Bodied Property

Expression-Bodied propertyler, kısmi olarak Auto Property Initializers'ın akrabasıdır diyebiliriz...

- Tanımlanan property'de Lambda Expression kullanmamızı sağlayan söz dizimidir.

```
public string Cinsiyet
{
    get
    {
        return "Erkek";
    }
}
```

```
public string Cinsiyet => "Erkek";
```

Bu şekilde expression-bodied
ile imzalanan propertyler
read only olarak
oluşturulacaktır.

Sonraki derslerimizde göreceğimiz Object initializer desteğidir.

```
Book book = new Book
{
    Author = "Kutsal İsyan",
    Name = "Hasan İzzet Dinamo"
};
```

Auto property-initializers object initializers'a izin vermemektedir..

```
Book book = new Book
{
    Author = "Kutsal İsyan",
    Name = "Hasan İzzet Dinamo"
};

book.Name = "Sabuncuoğlu Şerafettin";
```

Lakin bu özelliği Init-Only Properties desteklemekte ve sonrasında read only özelliği göstermektedir...

Init-Only Properties - Init Accessor (C# 9.0)

- Init-Only Properties, nesnenin sadece ilk yaratılış anında propertylerine değer atamaktadır.
- Böylece iş kuralı gereği run time'da değeri değişmemesi gereken nesneler için bir önlem alınmaktadır.

Init-Only properties, developer açısından süreç esnasında değiştirilmemesi gereken property değerlerinin "yanlışlıkla" değiştirilmesinin önüne geçmekte ve böylece olası hata ve bug'lardan yazılımı arındırmaktadır.

Böyle bir durumda aklınıza direkt Auto Property Initializers gelmiş olabilir...

O halde Init-Only Properties'in getirisi nedir?

Metot

- Nesne üzerinde, field'larda ki yahut dışarıdan parametreler eşliğinde gelen değerler üzerinde işlemler yapmamızı sağlayan temel programatik parçalardır.

Indexer

```
[erişim belirleyicisi] [geri dönüş değeri] this[ parametreler ]
{
        get {       }   → Indexer'dan veri istenildiğinde tetiklenir.

        set {       }   → Indexer'a veri gönderildiğinde tetiklenir.
                          Gönderilen veriyi value keywordüyle
                          yakalar.
}
```

- Nesneye indexer özelliği kazandıran, fıtrat olarak property ile birebir aynı olan elemandır.

Field

- Nesne içerisinde veri depoladığımız/tuttuğumuz alanlardır.