



# Object Oriented Programming

Destructor/Finalizer Functions



# Destructor/Finalizer Metot Nedir?

Bir class'tan üretilmiş olan nesne imha edilirken otomatik çağrılan metottur.

C# programlama dilinde Destructor sadece class yapılanmasında kullanılabilir ve bir class sade ve sadece bir adet Destructor içerebilir.

# Destructor Davranış Modeli

Selametele.....



# Peki Bir Nesne Hangi Şartlarda Kim Tarafından İmha Edilir?

- Bir nesnenin imha edilmesi için;
  - İlgili nesne herhangi bir referans tarafından işaretlenmemelidir,
  - Yahut nesnenin oluşturulduğu ve kullanıldığı scope sona ermiş olmalıdır.
  - Yani anlaşılan ilgili nesneye bir daha erişilemez hale gelinmelidir.

işte o zaman nesne imha edilir.

Peki kim  
tarafından?







# Garbage Collector

- Uygulamada lüzumsuz olan nesneleri toplamak için Garbage Collector isimli bir mekanizma devreye girer.
  - Esasında Garbage Collector C#'da bellek optimizasyonunu üstlenen bir yapılanmadır.
  - C#'da Garbage Collector'ın ne zaman iş göreceği tahmin edilemez. Kafasına göre takılır :)
    - Dolayısıyla biz geliştiricilerin bu mekanizmaya müdahale etmesi pek önerilmez!

# Destructor Tanımlama Kuralları

**class MyClass**

```
{  
  
    ~MyClass(){  
        //...işlemler...  
    }  
}
```

İşte bu fonksiyon, bu sınıftan üretilen nesne imha edilirken otomatik olarak son kez "selametle" demek için(yani son işlemleri yapabilmek için) tetiklenecek olan fonksiyondur.

Bir sınıf içerisinde sade ve sadece bir adet destructor tanımlanabilir.

Ve her özel sınıf elemanlarında olduğu gibi destructor'da sınıf ismiyle aynı isimde olan bir fonksiyondur.

Destructor tanımlayabilmek için ~(tilde) işareti kullanılır.

```
class MyClass  
{  
    1 reference  
    public MyClass() Constructor  
    {  
        Console.WriteLine("Selamın aleyküm...");  
    }  
    0 references  
    ~MyClass() Destructor  
    {  
        Console.WriteLine("Selametle...");  
    }  
}
```