Inheritance (Kalıtım)

Object Oriented Programming

Base Class ve D[illegible]ass Nedir?

Kalıtım veren sınıfa
Base / Parent Class
denir.

```
class Araba
{
    public string Marka { get; set; }
    public string Model { get; set; }
    public int KM { get; set; }
}
```

Kalıtım alan sınıfa
Derived / Child Class
Denir.

```
class Opel : Araba
{

}
```

Base Class ve Derived Class Kritik!

```
class A
{
   ...
}
class B : A
{
   ...
}
class C : B
{
   ...
}
class D : C
{
   ...
}
```

Base Class : C
Derived Class : D

Peki, atalar tümmm torunların Base Class'ı mıdır?

Yani A, B'nin olduğu gibi bir yandan da C ve D'nin de Base Class'ı mıdır?

Lakin atalarındaki tüm sınıflar Base Class'ı değildir!

Örneğin; C'nin Base Class'ı B'dir. A ise atasıdır lakin Base Class'ı değildir.

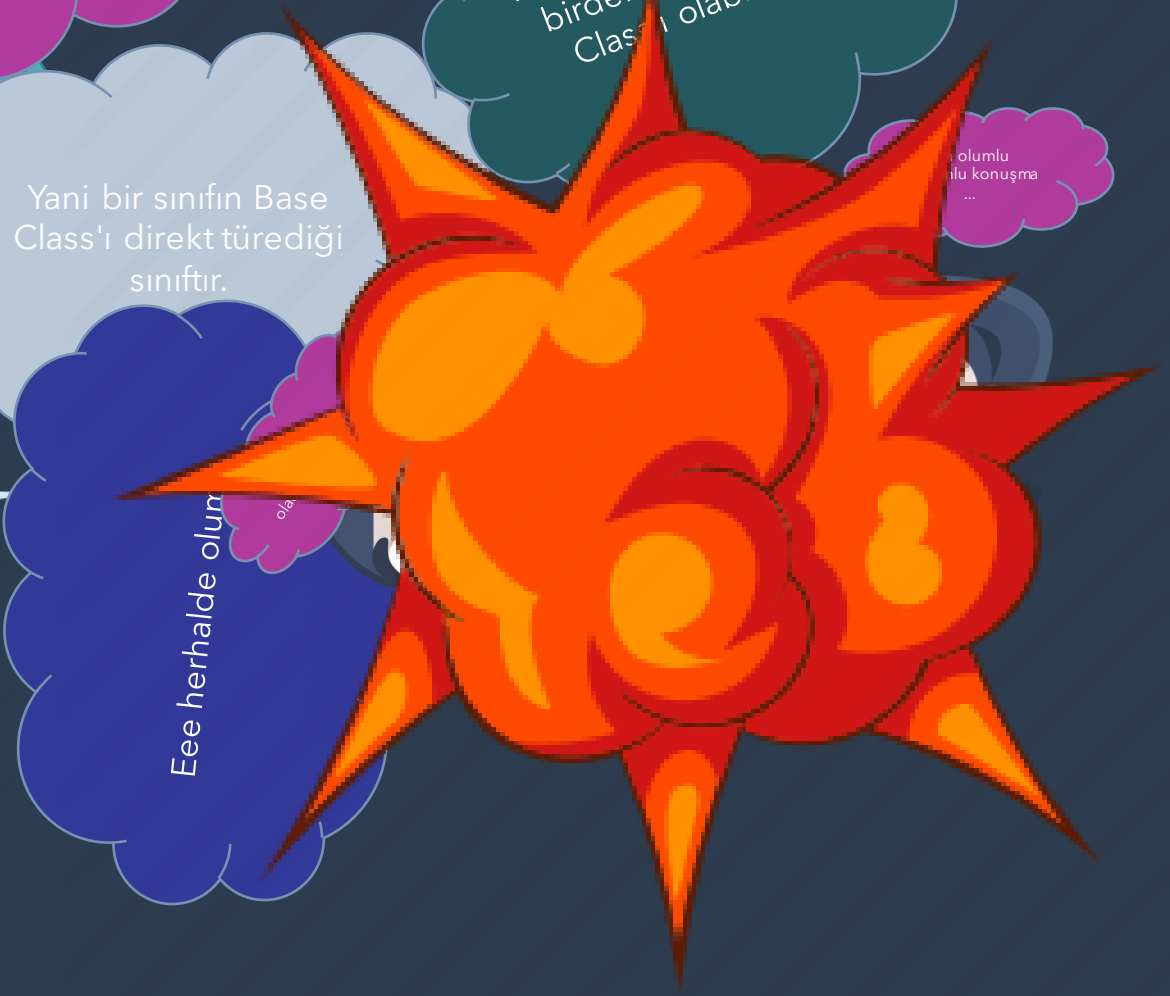
Unutma! Bir sınıfın sade ve sadece tek bir Base Class'ı olabilir!

Yani bir sınıfın Base Class'ı direkt türediği sınıftır.

Peki müdür, bir class'ın birden fazla Derived Class'ı olabilir mi?

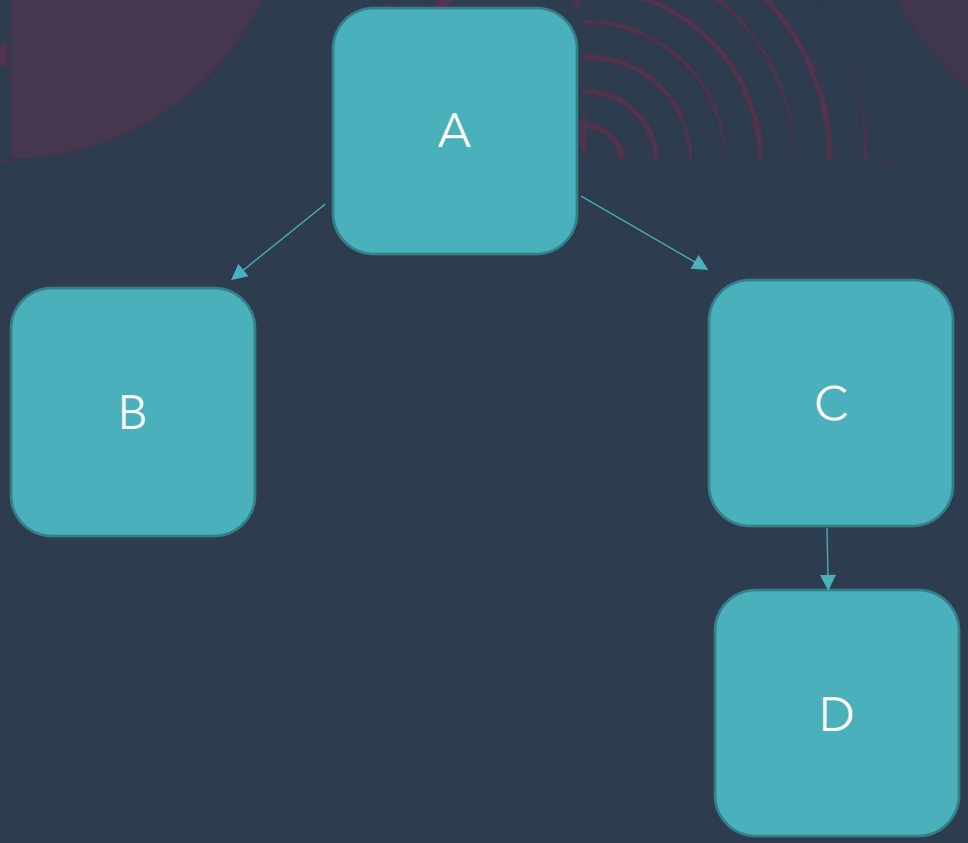
Eee herhalde olum

olumlu
lu konuşma
...



```
class A
{
    ...
}
class B : A
{
    ...
}
class C : B
{
    ...
}
class D : C
{
    ...
}
```

```
class A
{
   . . .
}
class B : A
{
   . . .
}
class C : A
{
   . . .
}
class D : C
{
   . . .
}
```



Kalıtımın Altın Kuralı!

- Bir class'ın sade ve sadece bir Base Class'ı olur dedik.
- Bunun nedeni, C# programlama dilinde bir class'ın sade ve sadece tek bir class'tan türetilmesine izin verilmektedir! Aynı anda birden fazla class'tan türeme işlemi gerçekleştirilemez!

```
class Y : X, Z, W...
{
    . . .
}
```

İleride bu şekilde birden fazla kalıtım tanımlamasının yapılabildiğini göreceksiniz.

Lakin orada da göreceksiniz ki Z ve W bir sınıf olmayacaktır!

(İpucu : Interface)

Kalıtımda Nesne Üretim Sırası

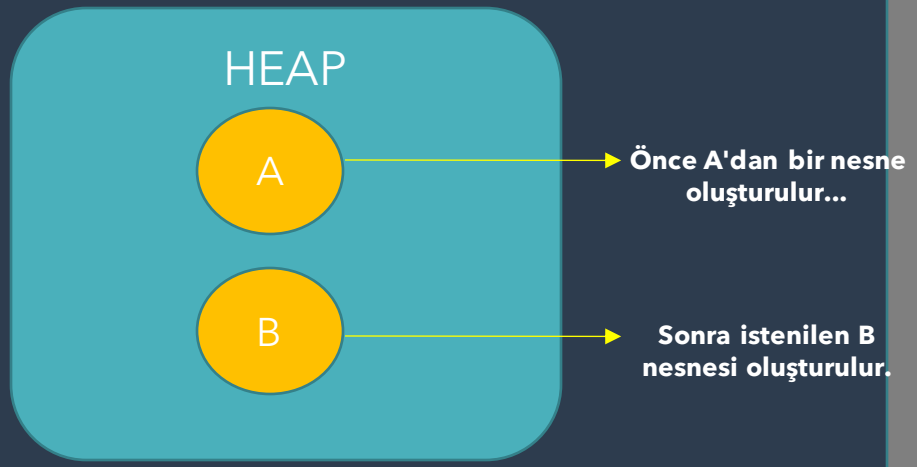
Yani buradan anlaşılıyor ki, bir sınıftan nesne üretilirken siz 1 adet nesne ürettiğinizi düşünsenizde kalıtımsal açıdan birden fazla nesne üretimi gerçekleştirilebilmektedir.

- Bir sınıftan nesne üretimi yapılırk
  önce o sınıflardan SIRAYLA nesn

Misal;

```
class A
{
   ...
}
class B : A
{
   ...
}
```

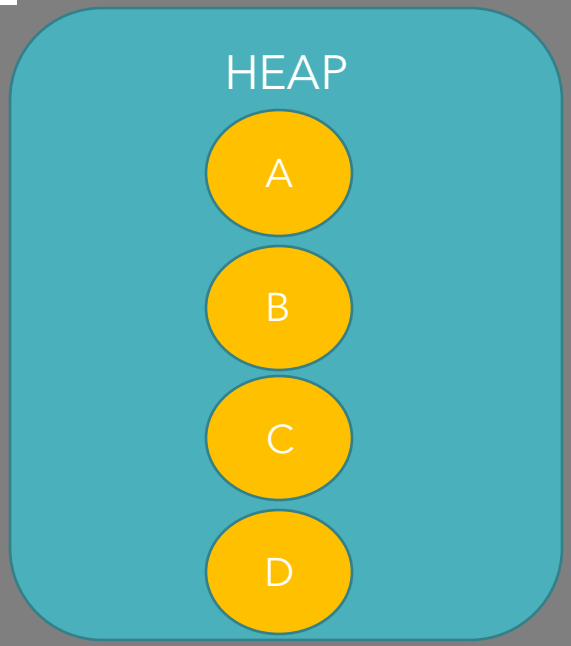
new B( ) → Komutu tetik...de



Başka Misal;

```
class A
{
   ...
}
class B : A
{
   ...
}
class C : B
{
   ...
}
class D : C
{
   ...
}
```

new D( )

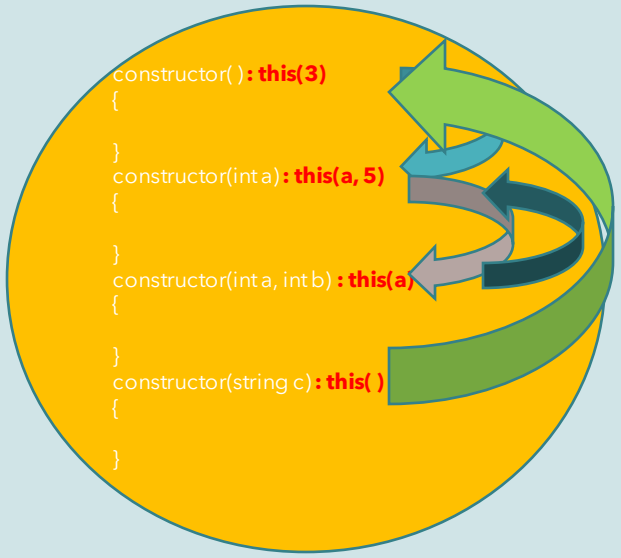
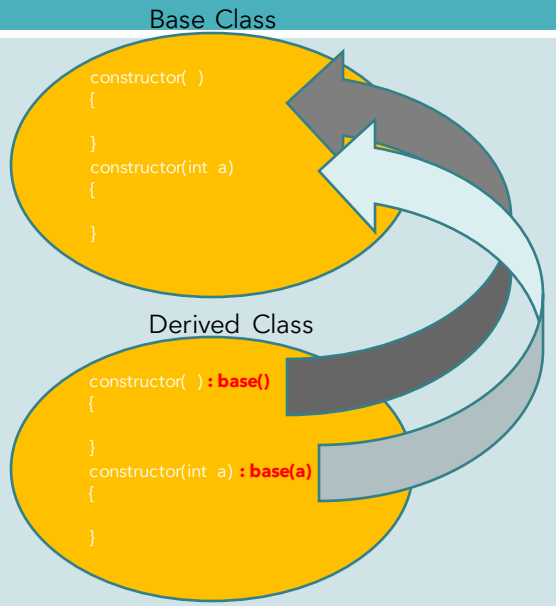


Bir Sınıftan Base Class Constructor'ına Ulaşım

- Madem ki, herhangi bir sınıftan nesne üretimi gerçekleştirilirken öncelikle base class'ından nesne üretiliyor, bu demektir ki önce base class'ın constructor'ı tetikleniyor.
- Haliyle bizler nesne üretimi esnasında base class'ta üretilecek olan nesnenin istediğimiz constructor'larını tetikleyebilmeli yahut varsa parametre bu değerleri verebilmeliyiz.
- İşte bunun için base Keyword'ü nü kullanmaktayız.

base Keyword vs this Keyword

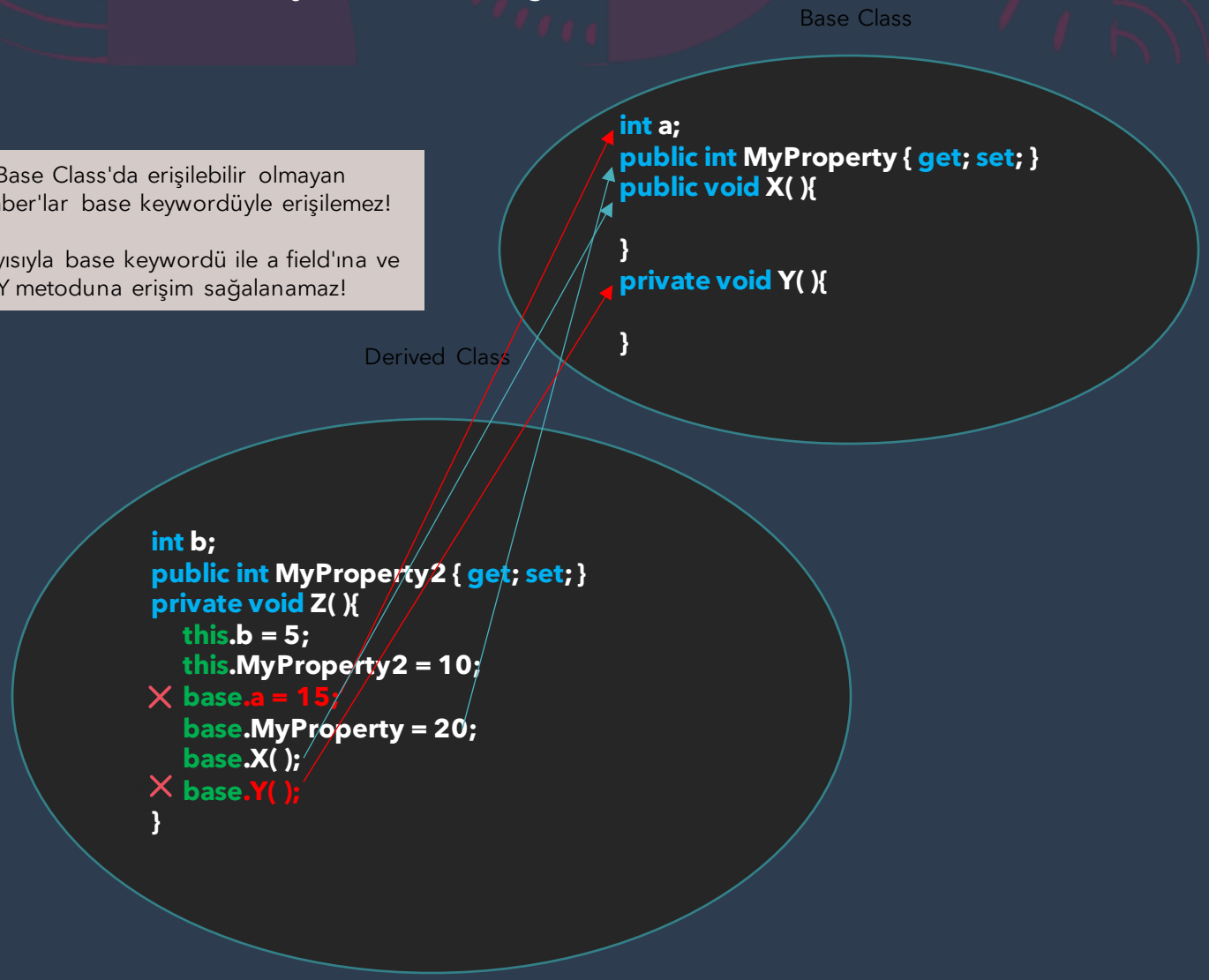
- this, bir sınıftaki constructor'lar arasında geçiş yapmamızı sağlar.
- base, bir sınıfın base class'ının constructor'larından hangisinin tetikleneceğini belirlememizi ve varsa parametrelerinin değerlerinin derived class'tan verilmesini sağlar.

this	base
	

Ayrıca nasıl ki this, ilgili sınıfta o anki nesnenin memberlarına erişebilmemizi sağlıyor, aynı şekilde base'de base class'da ki memberlara erişebilmemizi sağlamaktadır.

Base Class'da erişilebilir olmayan member'lar base keywordüyle erişilemez!

Dolayısıyla base keywordü ile a field'ına ve Y metoduna erişim sağalanamaz!



Yorulduk müdür!