



# **Bölüm 4: Çok Biçimlilik**

## **JAVA ile Nesne Yönelimli Programlama**



# Çok Biçimlilik (Polymorphism)

- Poly (many) morph (shape)
- Bir referans, aynı türde veya alt sınıfa ait bir nesneye atıfta bulunabilir.
- Nesne oluşturulurken, ilgili sınıfın yapıcı fonksiyonu çağrılır.

```
// Köpek nesnesine referans  
Kopek kopek = new Kopek();  
// Hayvan referansı, Kopek nesnesine referans olabilir  
Hayvan hayvan = kopek;  
// Object referansı, Hayvan nesnesine referans olabilir  
Object nesne = hayvan;
```



# Referans Kuralı

- Bir nesnenin alt türünü, o nesnenin üst sınıfının bir referansına atayabiliriz.
- Ancak, üst sınıfın nesnesini, alt sınıfın bir referansına atayamayız.

```
// Kopek, Hayvan'ın alt türüdür  
Hayvan hayvan = new Kopek(); // Geçerli  
// Hayvan, Kopek'in üst sınıfı  
Kopek kopek = new Hayvan(); // Hatalı
```



# Üst Sınıf ve Alt Sınıf

- Bir üst sınıf nesnesi, alt sınıflardan herhangi birinin nesnesi değildir.
- **is-a** ilişkisi yalnızca üst sınıflardan alt sınıflara yönelir.
  - Köpek **is a** Hayvan (her köpek bir hayvandır)
  - Hayvan **is not only a** Köpek (her hayvan bir köpek değildir)



# Tür Dönüşümü (Downcasting)

- Bir üst sınıf referansı, alt sınıf türüne açıkça (explicitly) dönüştürülebilir.
- Üst sınıf referansının alt sınıf değişkenine atanmasına izin verilir.
- Downcasting üst sınıfa ait olmayan metotların çağrılması için kullanılır.
- Bir fonksiyon çağrısında, aramaya hiyerarşinin en altından başlanır.



# Dinamik Bağlama (Dynamic Binding)

- Bir nesne, alt sınıfa referans ise, alt sınıfın metotları çağrılır.
- Çağrılacak metot, değişkenin sınıf türüne bağlı olarak belirlenir.
- Bu işlem *dinamik bağlama* olarak adlandırılır.

```
Hayvan hayvan = new Kopek();  
hayvan.kos(); // Kopek sınıfının fonksiyonu çağrılır.
```



# Kalıtım ve Çok Biçimlilik

- Kalıtım, sınıf hiyerarşilerini oluşturmak için gereklidir.
- Çok biçimlilik için kalıtım gereklidir. (Sınıf hiyerarşilerine sahip olmalıyız)
- Kalıtımı sağlamak için çok biçimlilik gerekli değildir.
- A sınıfının nesnesi, B sınıfının nesnesi gibi davranabilir. (A, B'nin üstüdür)
- Üst sınıf nesnesini alt sınıf referansına atamak mümkündür.
- Bu işleme daraltma dönüşümü denir. (casting)
- Parantez kullanılarak yapılır.
- `B objB = (B) objA; // Legal, ancak dikkatli kullanılmalı`



SON