



**BTK**  
**AKADEMİ**

# Nesne Yönelimli Programlama

Zafer CÖMERT



# Bölüm 5

## Kalıtım (Inheritance)

## Bölüm 5

### Kalıtım (Inheritance)

- Kalıtımın türleri ve uygulanması
- Sınıf hiyerarşisi
- Metot gizleme ve geçersiz kılma
- virtual – override
- Çokbiçimlilik (Polymorphism)

1

2

3

4

5

6

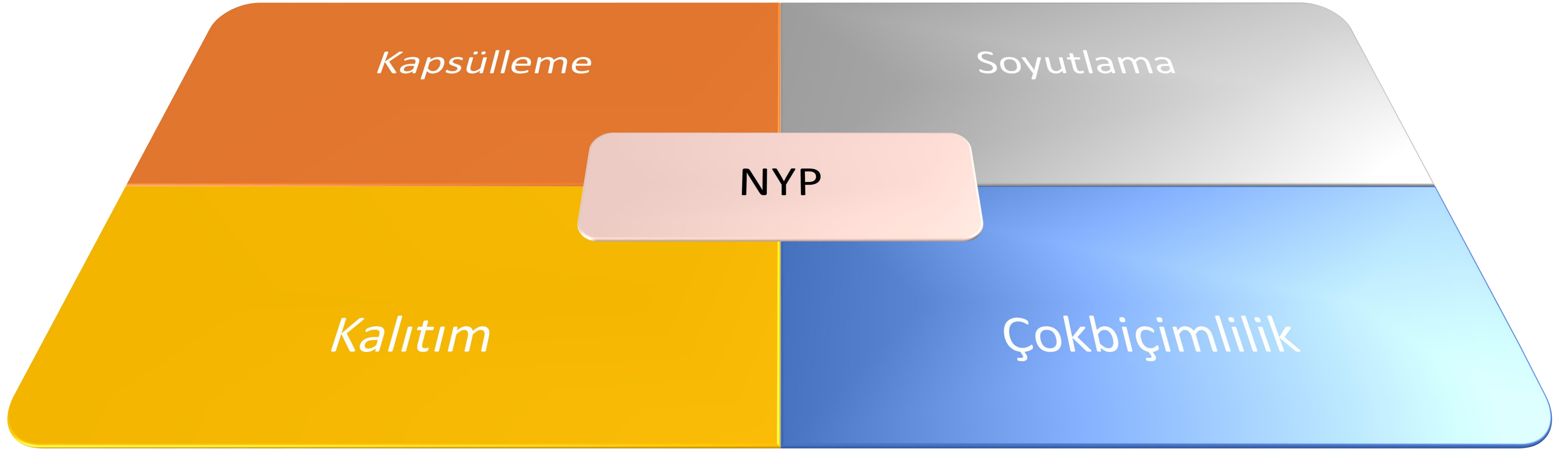
7

8

9

10

# NYP'nin Karakteristik Özellikleri



# Kalıtım (Inheritance)

- Herhangi bir kodu tekrar kullanabilmekten bahsettiğimizde genellikle kalıttan bahsederiz. Kalıtım, bir nesnenin başka bir nesnenin özelliklerini devralması işlemidir.
- Kalıtım, bir sınıfın özelliklerinin, davranışlarının ve yöntemlerinin başka bir sınıf tarafından devralınabilmesini sağlar.
- Böylece, yeni bir sınıf, bir önceki sınıfın tüm özelliklerini ve davranışlarını miras alır ve bunları üzerinde değişiklikler yapabilir.



# Kalıtım (Inheritance)

Single  
Inheritance

Multi-level  
Inheritance

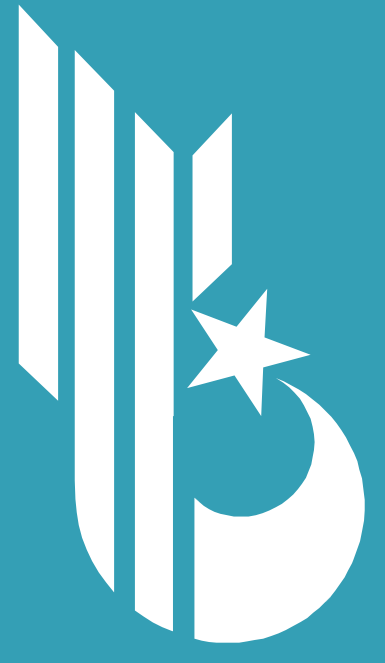
Hierarchical  
Inheritance

Multiple  
Inheritance

Multiple and  
Multilevel  
Inheritance

Interface  
Inheritance

Virtual  
Inheritance

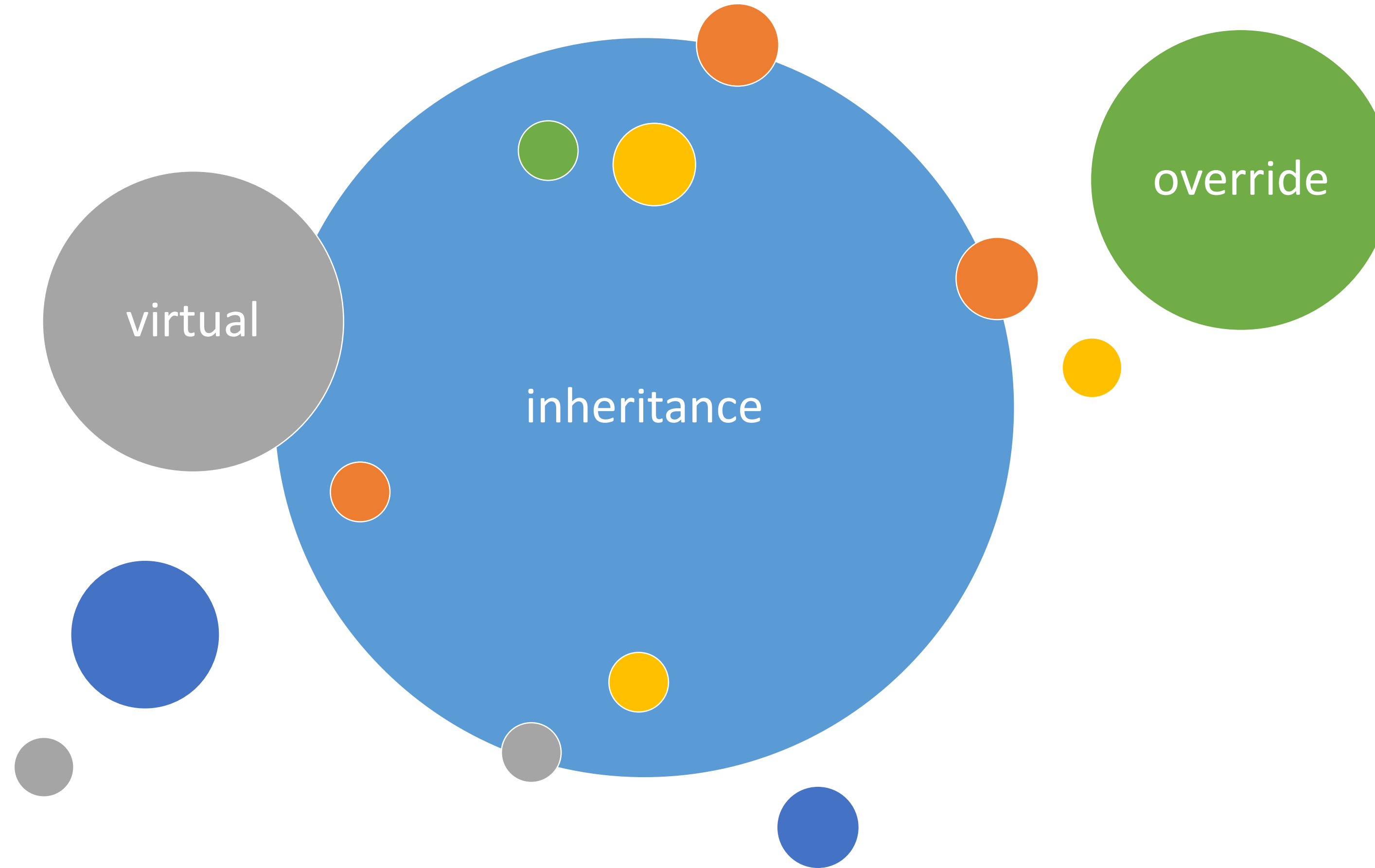


**BTK**  
**AKADEMİ**

*Kalıtım, bir sınıfın özelliklerinin,  
davranışlarının ve yöntemlerinin  
başka bir sınıf tarafından  
devralınabilmesini sağlar.*

*Böylece, yeni bir sınıf, bir önceki sınıfın tüm özelliklerini ve davranışlarını miras alır ve bunları üzerinde değişiklikler yapabilir.*

# Kalıtım (Inheritance)





# Çokbiçimlilik (Polymorphism)

- Polymorphism, bir nesnenin birden fazla farklı şekilde davranış sergileyebilmesi yeteneğini ifade eder.
- Bu davranışlar, aynı isimle ancak farklı parametrelerle çağrılan farklı metotlar tarafından gerçekleştirilir.
- Polymorphism, bir arayüz üzerinden birden fazla farklı sınıfın uygulanmasına izin verir. Bu sayede, aynı işlevselliğe sahip olan ancak farklı davranışlar sergileyen nesneler birbirinin yerine kullanılabilir hale gelir.

# C++



- Nesne Yönelimli Programlama (OOP)
- Generic Programlama
- Yapısal Programlama
- Fonksiyonel Programlama (FP)
- Asenkron Programlama