



BTK
AKADEMİ

Nesne Yönelimli Programlama

Zafer CÖMERT



Bölüm 4

Sınıf Tanımı, Üyeleri ve Yapısı

Bölüm 4

Sınıf Tanımı, Üyeleri ve Yapısı

- Sınıf Tanımlama
- Metotlar
- Yapıcı metot
- Nesne türetme / nesne başlatma
- Kapsülleme
- Soyutlama

1

2

3

4

5

6

7

8

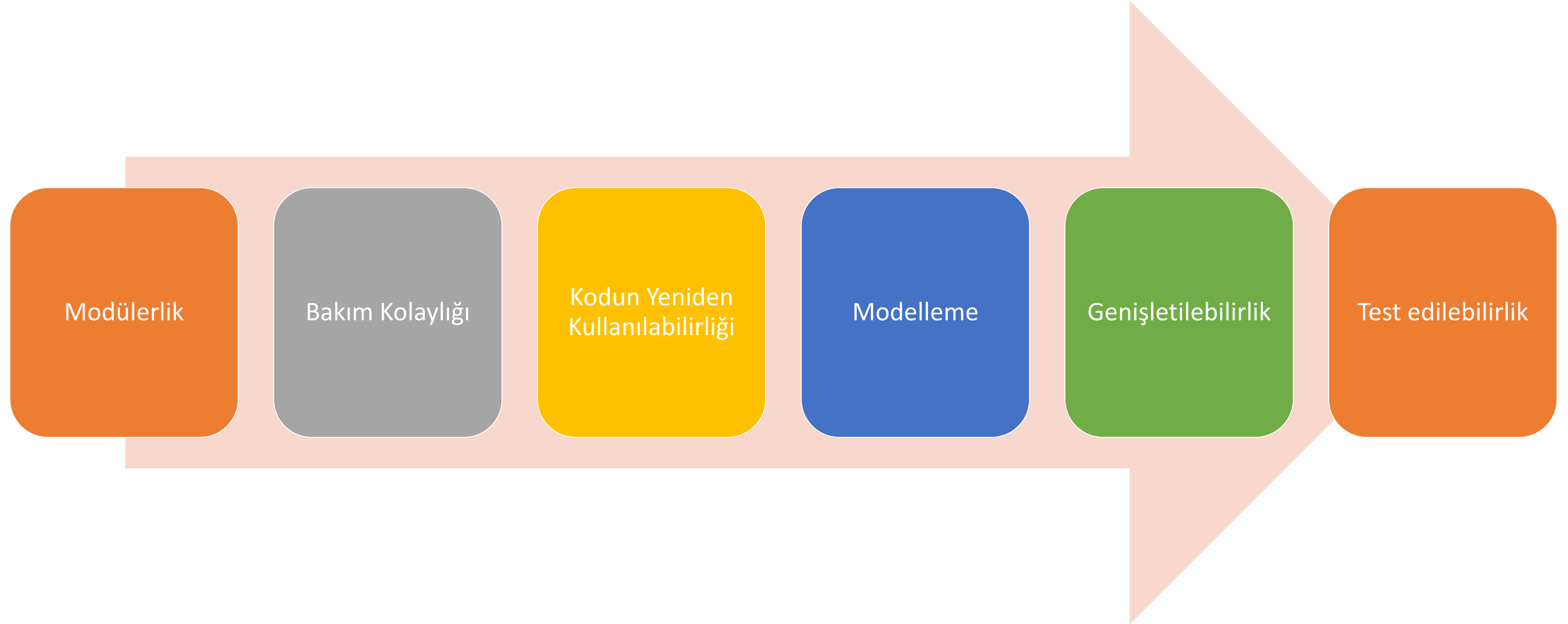
9

10

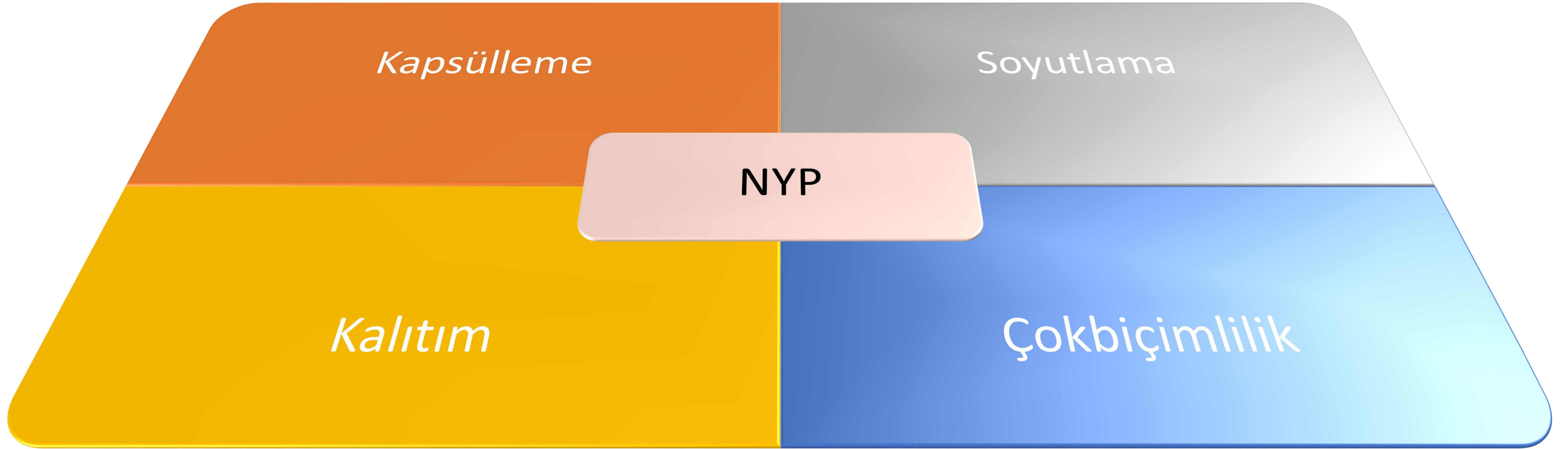
NYP Giriş

- Nesne yönelimli programlama (NYP) paradigmasının temel motivasyonu, yazılım tasarımının **gerçek dünya nesnelere** daha uygun hale getirilmesidir.
- Bu paradigma, gerçek dünyadaki nesnelerin **özelliklerini ve davranışlarını** yazılım tasarımına yansıtmayı hedefler.

NYP Neden Kullanılır?



NYP'nin Karakteristik Özellikleri



Nesneler ve Sınıflar

- NYP, sınıflar ve nesneler gibi temel kavramlara dayanır.
- Sınıflar, nesnelerin yapısını tanımlamak için kullanılan taslak veya şablonlardır.
- Nesneler ise, sınıfların örnekleridir ve sınıftaki özellikleri ve davranışları taşırlar.

Nesne



Object

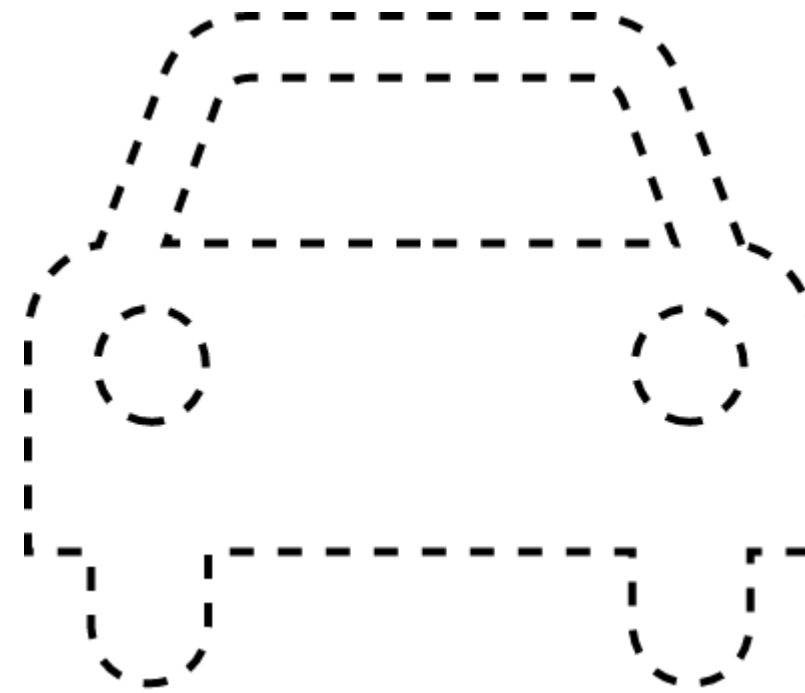


Object



Object - Class

- Sınıflar, nesnelerin yapısını tanımlamak için kullanılan taslak (**template**) veya şablonlardır (**blueprint**).



Mantıksal bir varlık

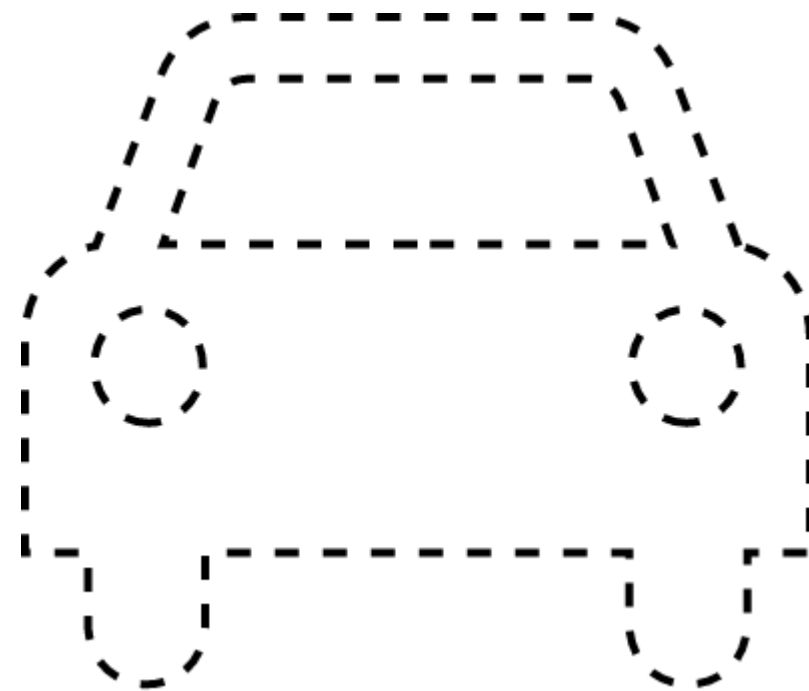
Object - Class

- Nesneler ise, sınıfların örnekleridir (**instance**) ve sınıftaki özellikleri ve davranışları taşırlar.



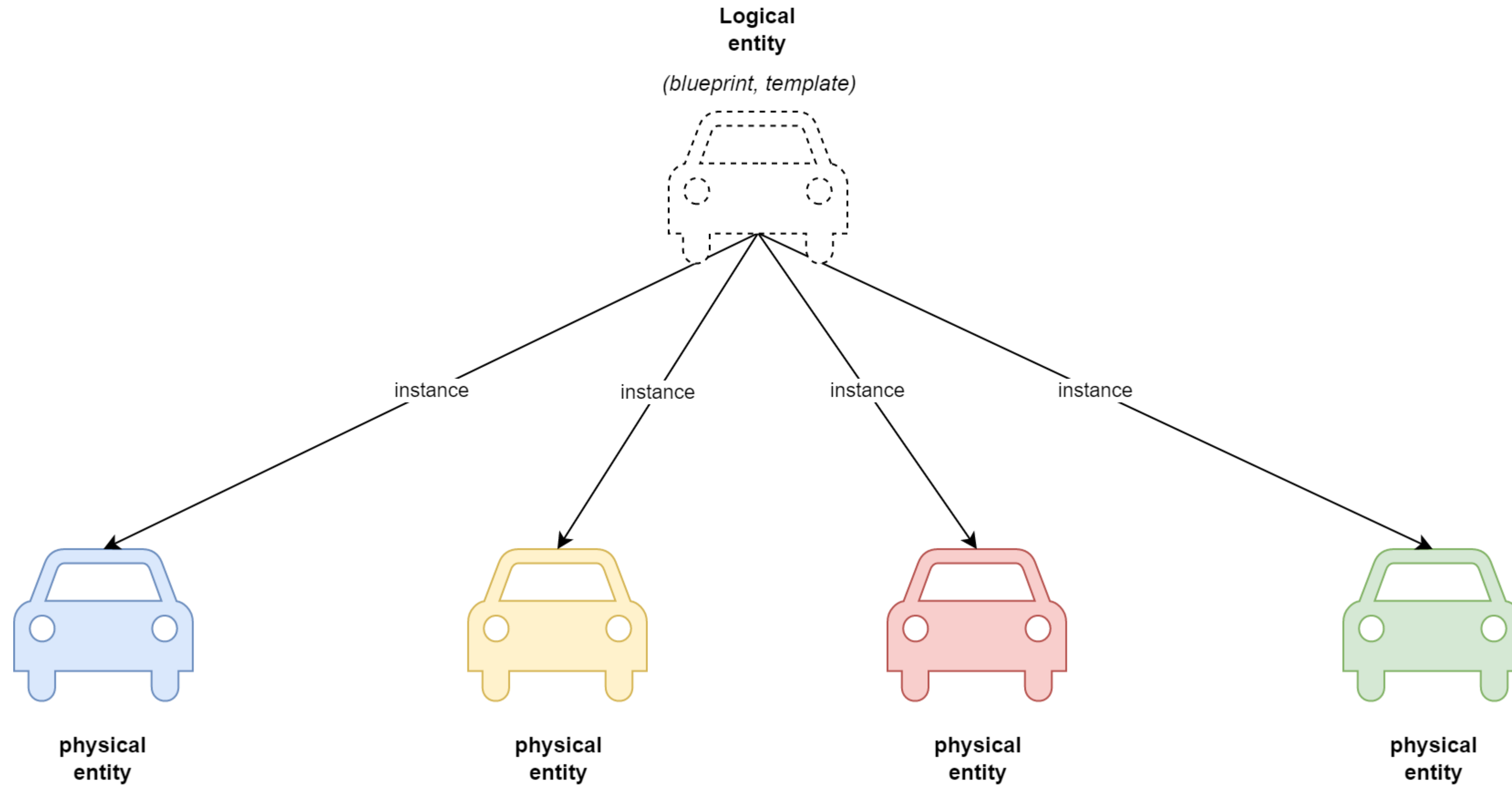
Fiziksel bir varlık

Object - Class

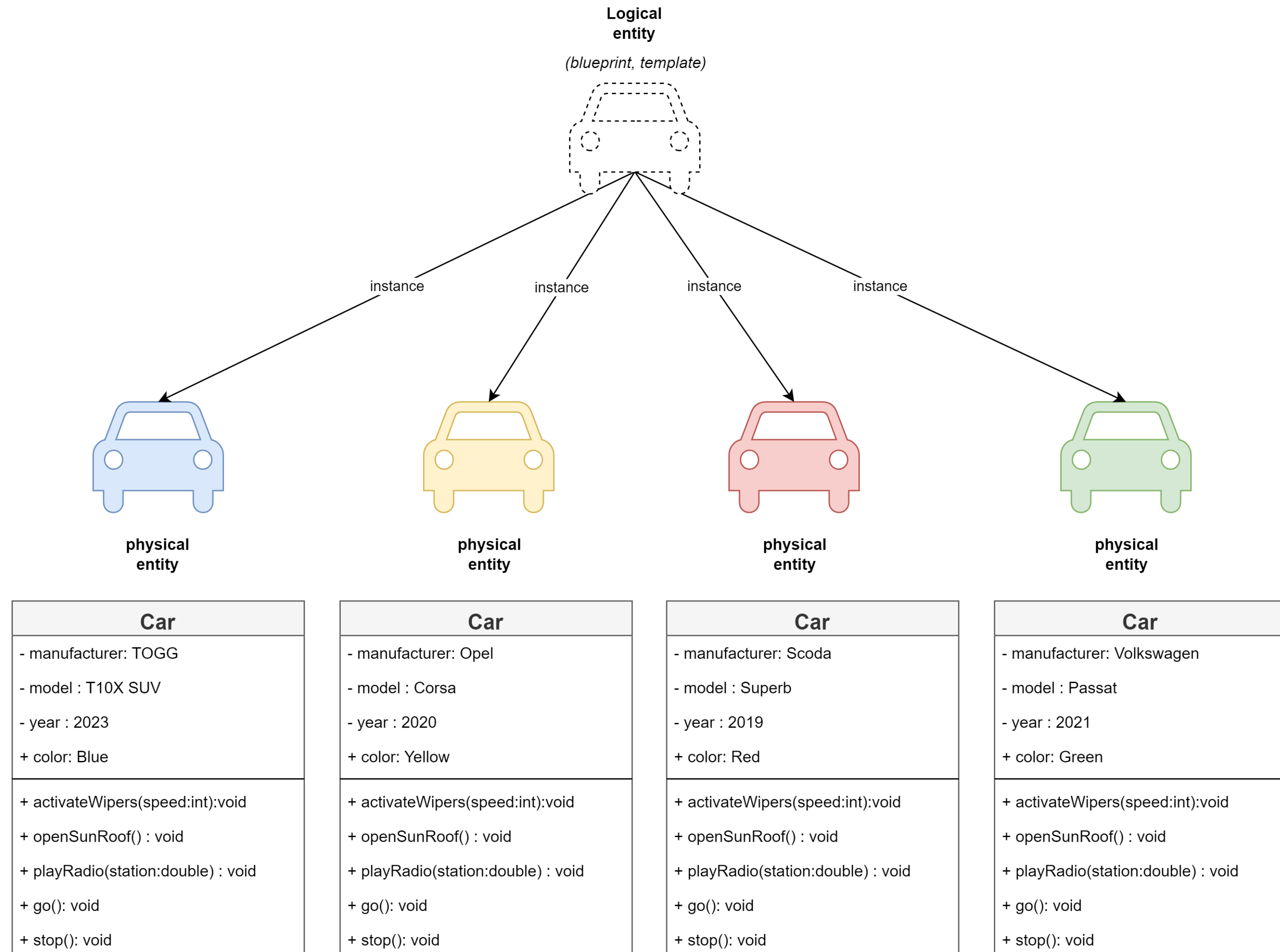


Car
<ul style="list-style-type: none">- manufacturer: string- model : string- year : string+ color: string
<ul style="list-style-type: none">+ activateWipers(speed:int):void+ openSunRoof() : void+ playRadio(station:double) : void+ go(): void+ stop(): void

Object - Class



Object – Class - Instance



Python



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

Java



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

Soyutlama (Abstraction)

- Soyutlama, bir nesnenin yalnızca önemli olan özelliklerinin gösterilmesini sağlar.
- Bir sınıf, bir nesnenin önemli özelliklerini tanımlar ve bu özellikleri diğer nesnelerden gizler.
- Bu, nesnelerin daha az karmaşık hale gelmesini ve programcının kodu daha iyi anlamasına yardımcı olur.

Soyutlama (Abstraction)

- Kahve makinesi bir nesne olarak düşünelim. Dış dünyadan sadece bazı işlevleri görebiliriz (örneğin, düğmeye basarak kahve yapabiliriz), ancak içindeki karmaşık mekanizma ve yapı gizlenir.
- Kullanıcılar sadece kahve yapma işlevini düşünür, içindeki detaylarla uğraşmak zorunda kalmazlar. Bu aslında soyutlama konusunda bir



Soyutlama (Abstraction)

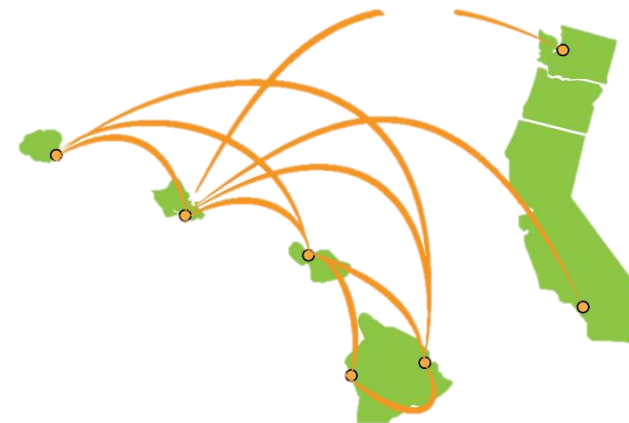
- TV kumandası, televizyonu açma, kanal değiştirme, ses ayarlama gibi işlevleri kullanıcıya sunar.
- Ancak, kumanda içindeki devreler ve sinyal iletimi gibi detaylar kapsülendir ve kullanıcı tarafından düşünülmez.



Soyutlama (Abstraction)



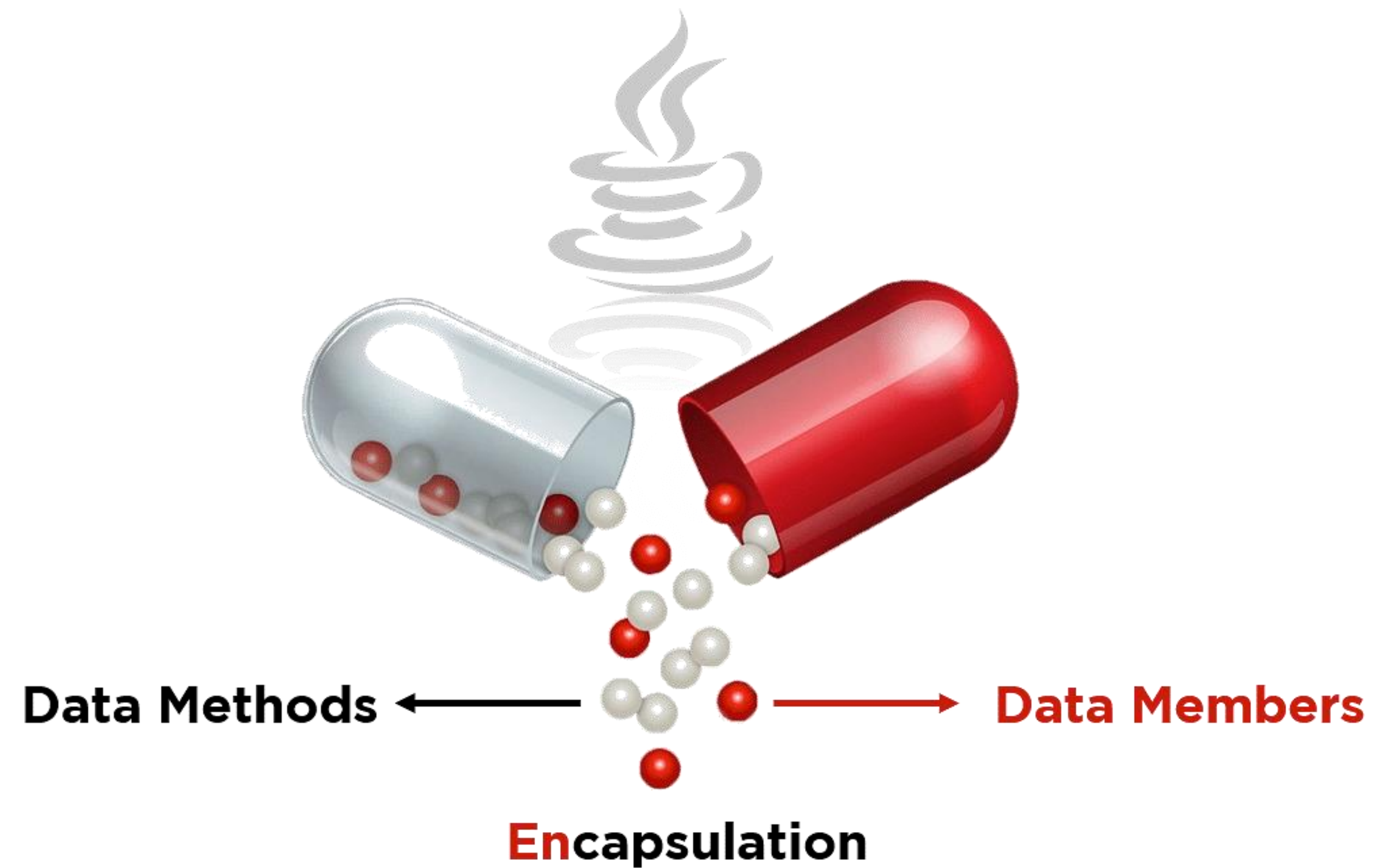
Soyutlama (Abstraction)



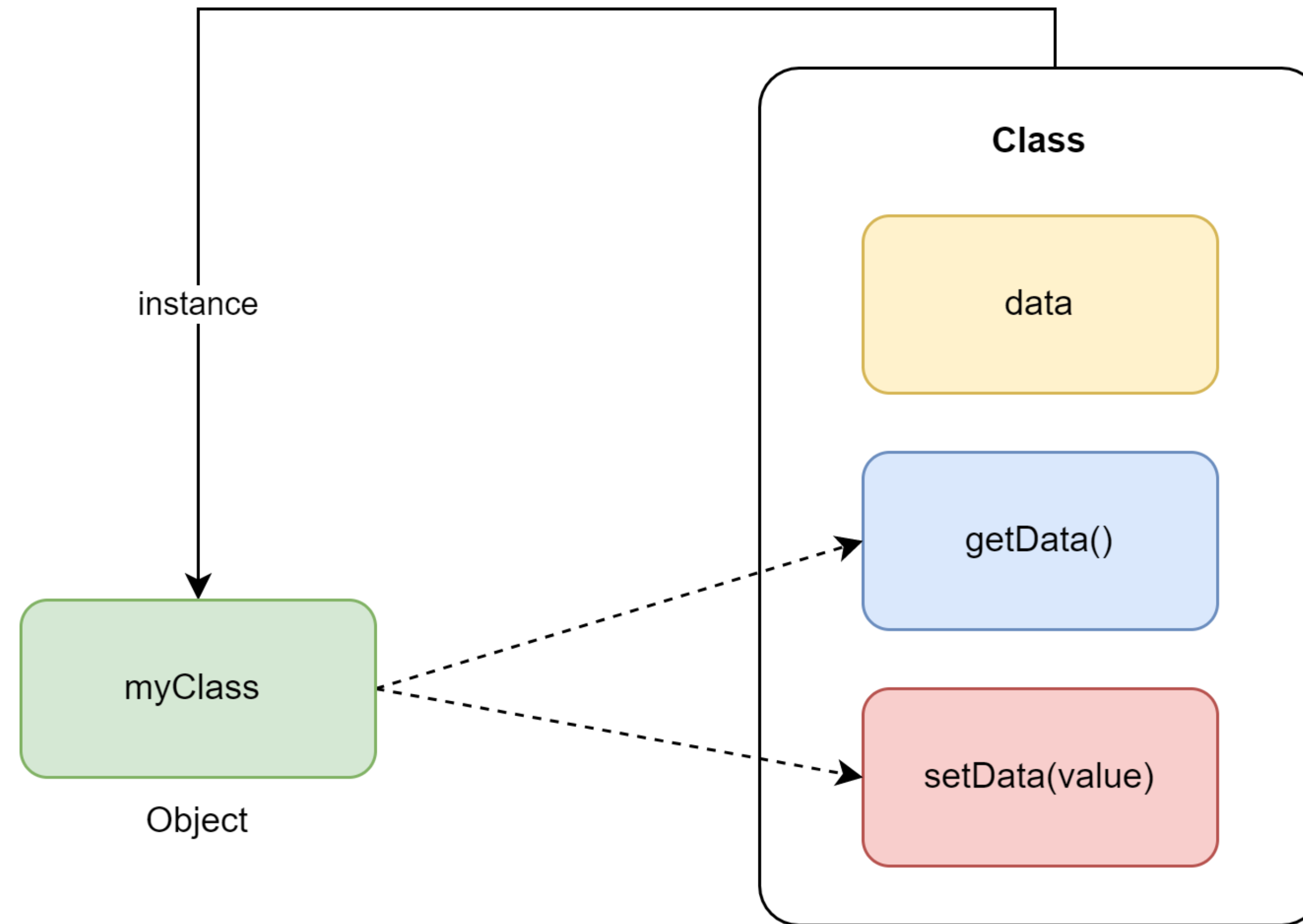
Kapsülleme (Encapsulation)

- Bu özellik, programcıların kodunun belirli bir bölümünü saklamasına (**information hiding**) ve programın diğer bölümleri tarafından daha az etkileşimli hale getirmesine yardımcı olur.
- Bu, programcının daha az hata yapmasına ve kodu daha kolay okunabilir ve anlaşılabilir hale getirmesine yardımcı olur.
- Kapsülleme, verilere doğrudan erişim izni verilmeyen, bunun yerine verilerin gizlendiği bir süreçtir.

Kapsülleme (Encapsulation)

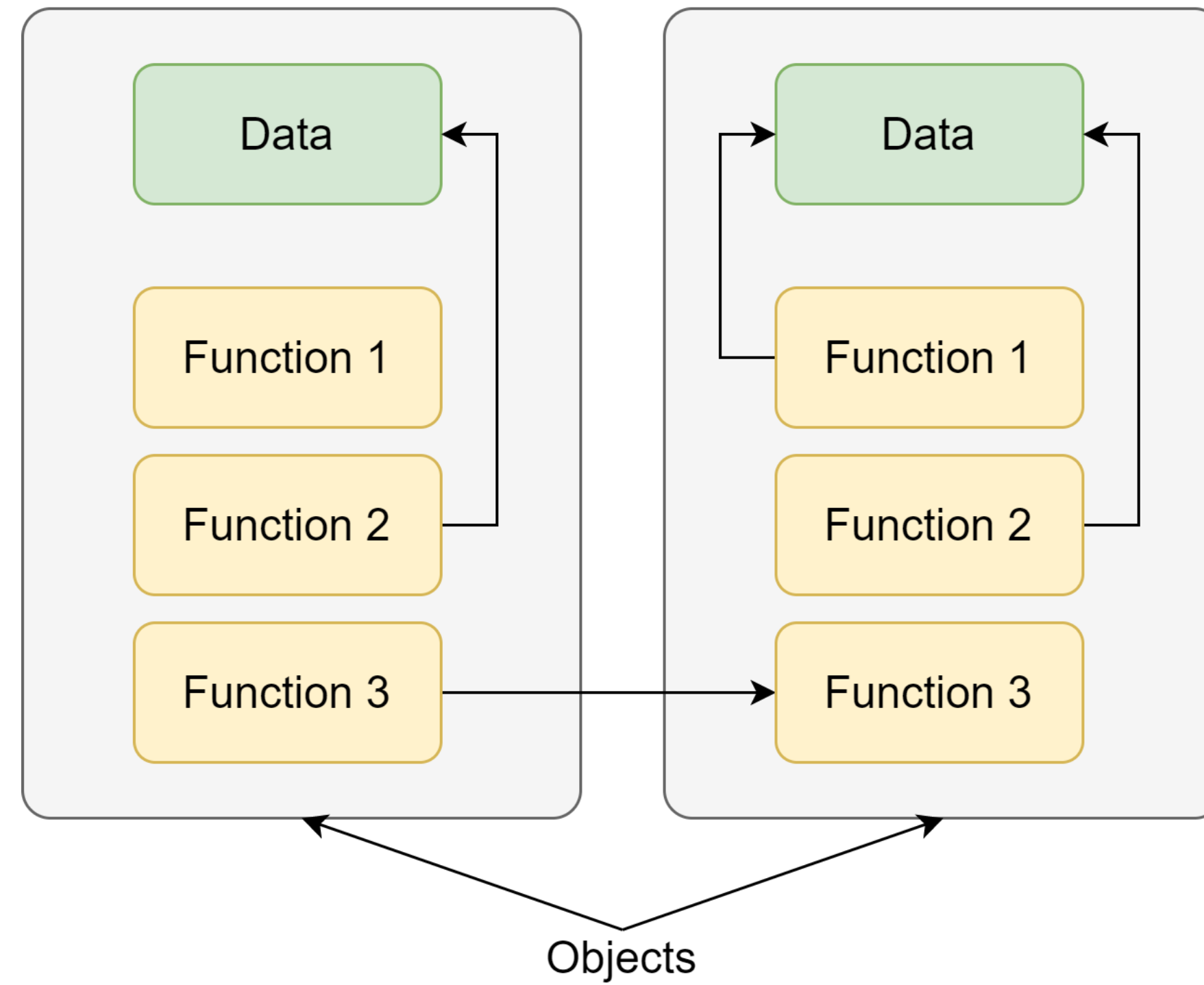


Kapsülleme (Encapsulation)



Kapsülleme (Encapsulation)

Nesne Yönelimli Programlama



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.

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.