Tasarım Örüntüleri
Chain of Responsibility

Örüntülerin Temel Prensipleri

- GoF tasarım örüntülerinin altında yatan temel prensipler
  - Encapsulation
  - Composition
  - Abstract Data Types

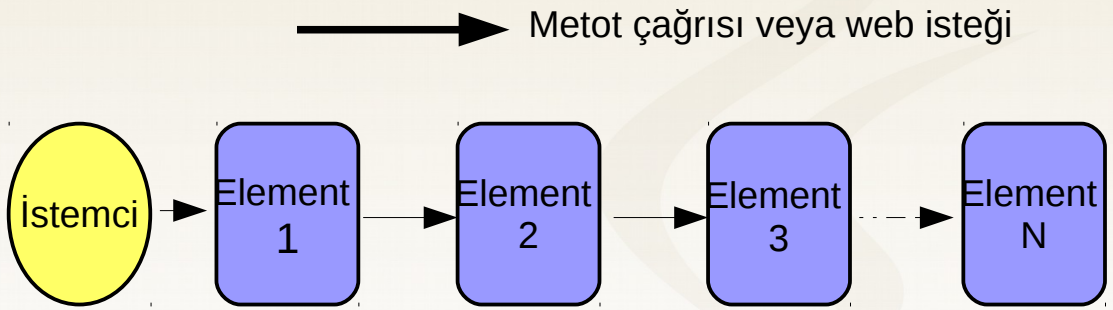
Chain of Responsibility Örüntüsü

- Bir **metot çağrısı**nın bazı durumlarda **birden fazla nesne**ye gönderilmesi gerekebilir
- Bu metot çağrısının tam olarak **hangi nesne tarafından** ele alınacağı önceden bilinememektedir
- Metot çağrısının veya isteğin **birden fazla nesne tarafından** ele alınması da söz konusu olabilir

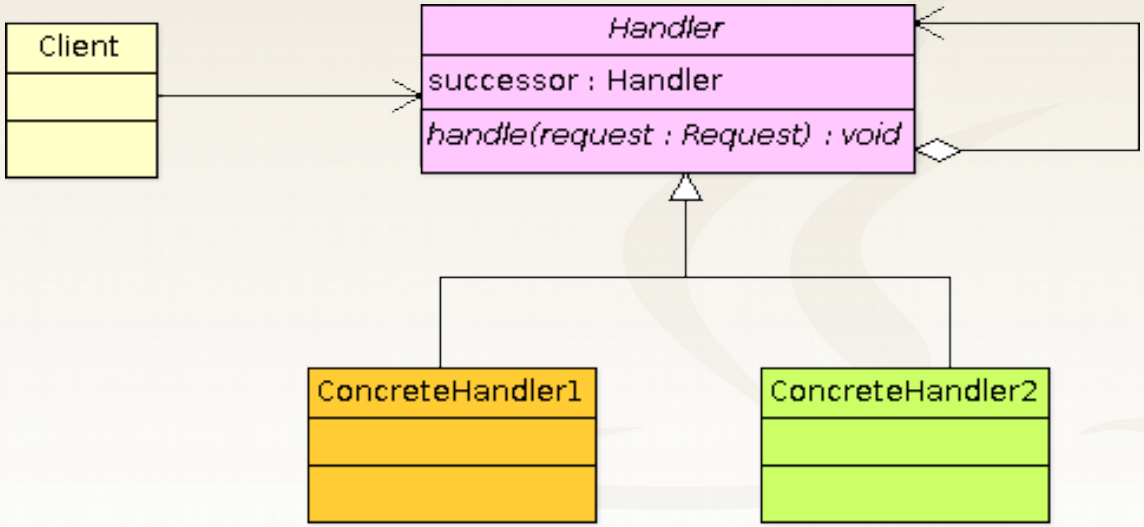
Chain of Responsibility Örüntüsü

- Bu gibi durumlarda metot çağrısını veya isteği ele alabilecek nesnelerden oluşan **bir nesne zinciri** oluşturulabilir
- İstek bu zincir üzerinde **bir nesneden diğerine iletilerek** ilerler

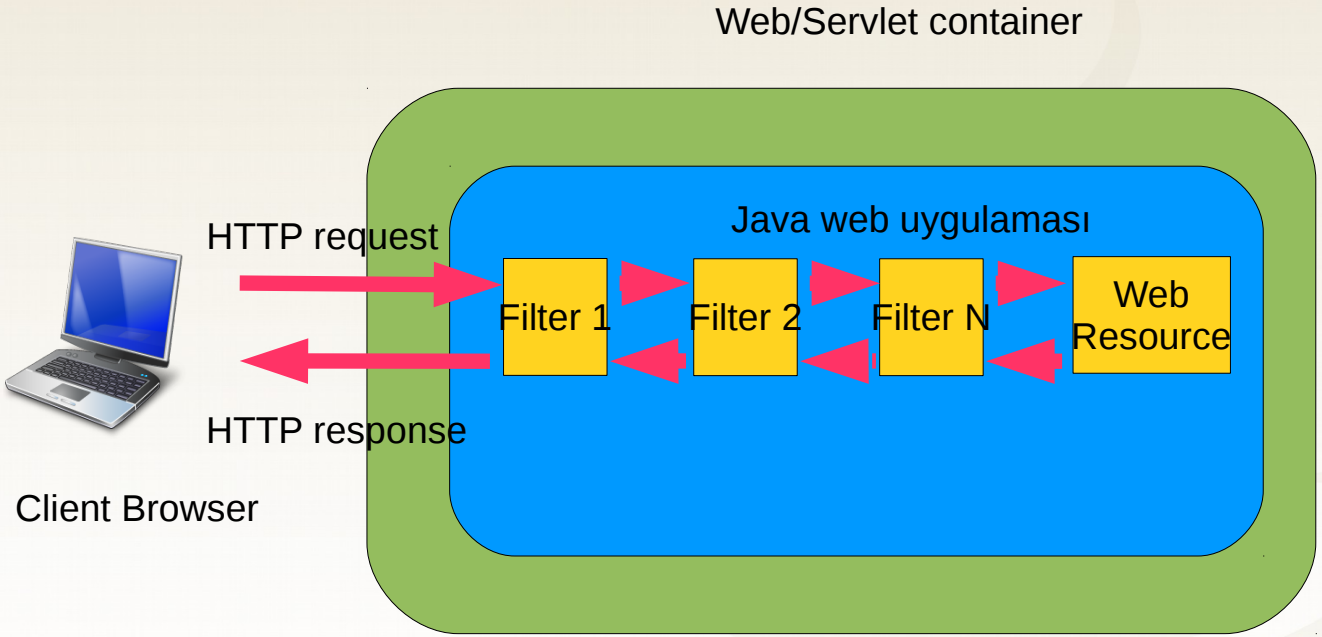
Chain of Responsibility Örüntüsü



Chain of Responsibility Örüntüsü Sınıf Diagramı



Chain of Responsibility Örüntüsünün Java İçerisinde Kullanımı



Chain of Responsibility Örüntüsünün Spring İçerisinde Kullanımı

- Spring MVC Framework içerisinde **MVC Interceptor kabiliyeti** de bir chain of responsibility gerçekleştirimidir
- **Controller endpoint'leri**ne gelen web isteklerinin intercept edilip **istek öncesinde ve sonrasında** bir takım işlemler yapmaya olanak tanırlar

Chain of Responsibility Örüntüsünün Sonuçları

- İsteğe cevap verecek **nesnelerin birbirlerini tanımaları** gerekmez
- Nesneler arasındaki ilişki daha basittir, **sadece bir sonraki nesne** bilinir
- **Sorumluluklar** zincirdeki **farklı nesnelere** dağıtılmaktadır

İletişim

JAVA Eğitimleri

Kurumsal Java Eğitimleri

www.java-egitimleri.com

info@java-egitimleri.com

@javaegitimleri

youtube.com/c/
KurumsalJavaEğitimleri