

# Het Chain of Responsibility Pattern



Objectgeoriënteerde analyse  
en ontwerp

# Het Chain of Responsibility Pattern

## Doel

Het object dat een verzoek doet aan een ander object loskoppelen van de concrete ontvanger.

Concrete ontvangers kunnen het verzoek doorgeven aan elkaar.

# Het Chain of Responsibility Pattern

## Deelnemers

- **Client:** stuurt het verzoek, naar een abstracte Handler.
- **Handler:** interface die de Client en concrete handlers van elkaar loskoppelt.  
Elke Handler heeft een verwijzing naar een *opvolger*.
- **ConcreteHandler:** een concrete implementatie van een Handler

# Chain of Responsibility: voordelen

- + Client moet concrete handler niet kennen.
- + Meerdere handlers mogelijk
- + Dynamische configuratie van keten met handlers mogelijk

# Chain of Responsibility : nadelen

- Geen garantie dat een verzoek uitgevoerd wordt
- Ketting van handlers moet goed geconfigureerd zijn