

# Softwaretechnik 1 - 4. Tutorium

Tutorium 03

Felix Bachmann | 26.06.2017

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



1 Orga

2 Recap

3 Stellvertreter

4 Vermittler

5 Tipps

## Ansage der Übungsleiter

- ab jetzt keine Abgabe per Mail mehr!  
⇒ auch nicht in Ausnahmefällen

# 4. Übungsblatt Statistik

## Allgemein



## Aufgabe 1 (Zustandsdiagramm - LEZ): $\emptyset$



## Aufgabe 2 (Abbottsche Methode): Ø



## Aufgabe 3 (iMage-GUI): Ø



## Aufgabe 4 (Geheimnisprinzip): Ø



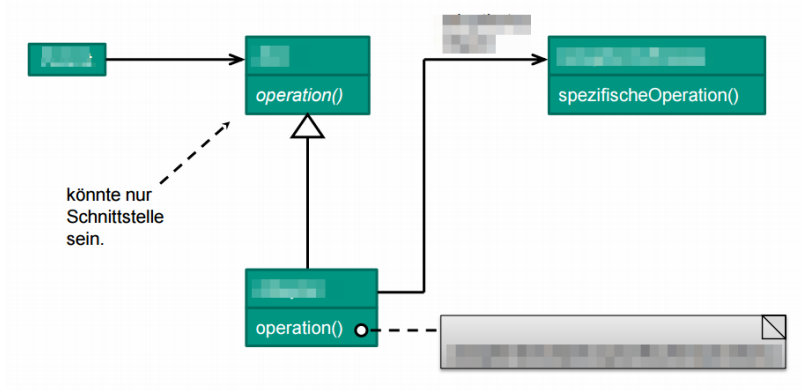


# Was bisher geschah..

- haben uns erste Entkopplungsmuster angeschaut

# Was bisher geschah..

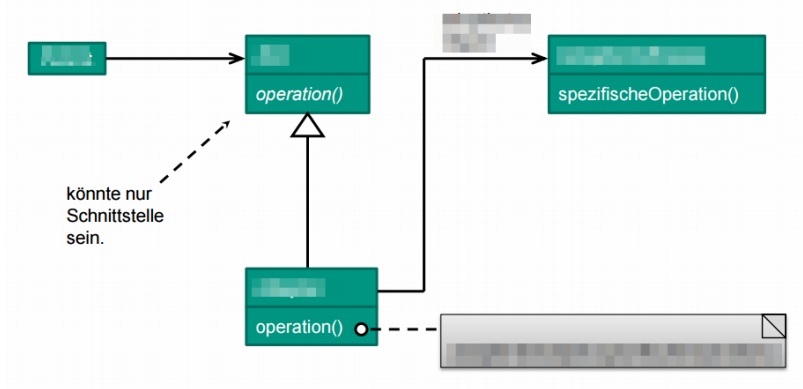
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



Welches Entwurfsmuster?

# Was bisher geschah..

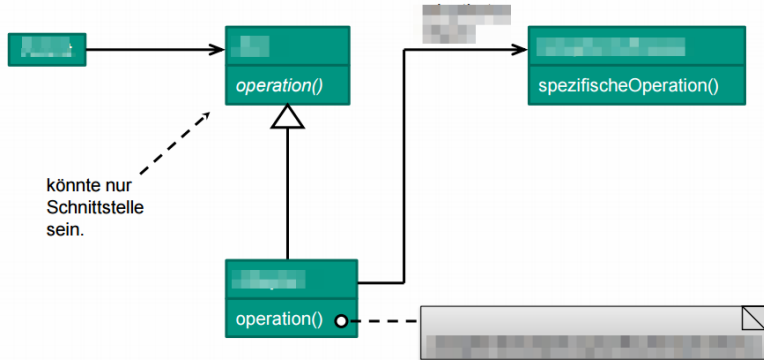
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



Welches Entwurfsmuster? (Objekt-)Adapter

# Was bisher geschah..

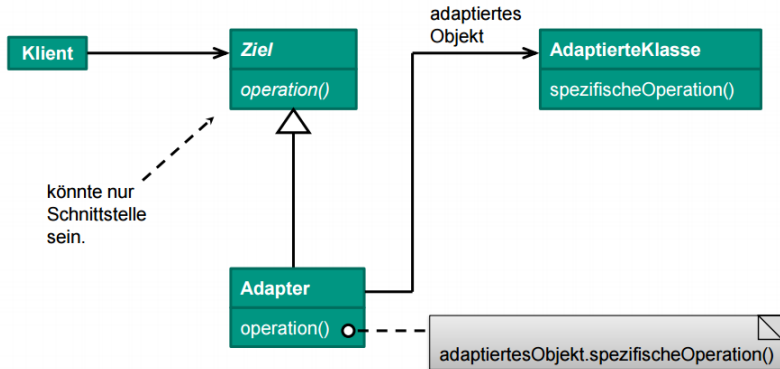
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



## Welche Klassen?

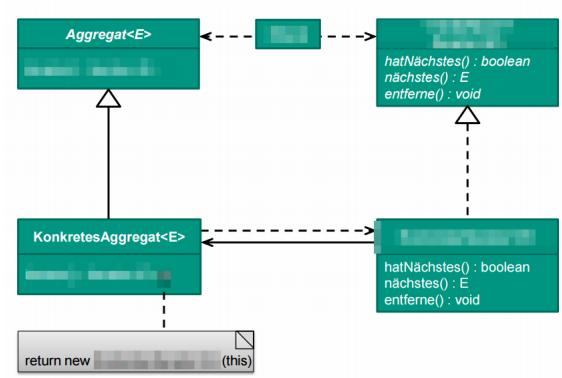
# Was bisher geschah..

- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



# Was bisher geschah..

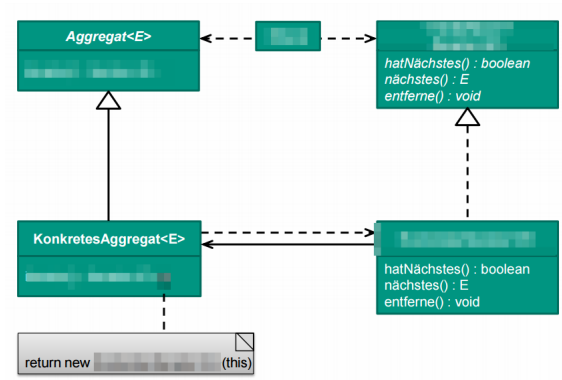
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



## Welches Entwurfsmuster?

# Was bisher geschah..

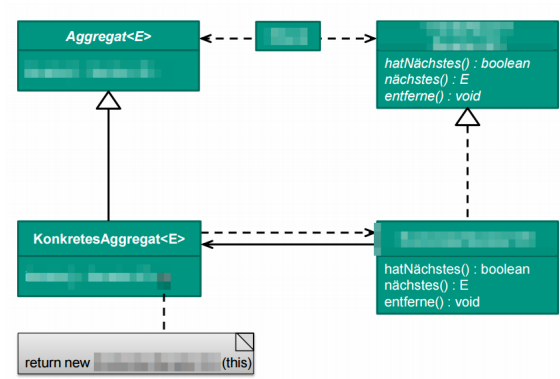
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



## Welches Entwurfsmuster? Iterator

# Was bisher geschah..

- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter

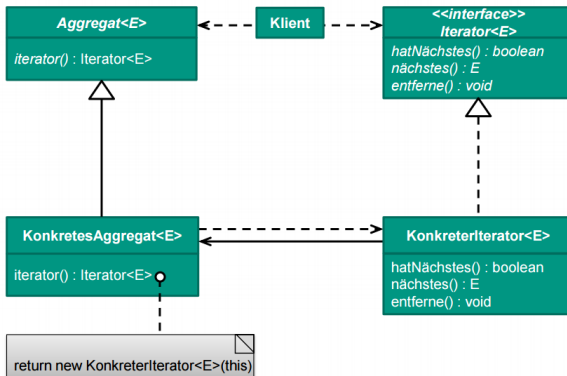


## Welche Klassen und Methoden?



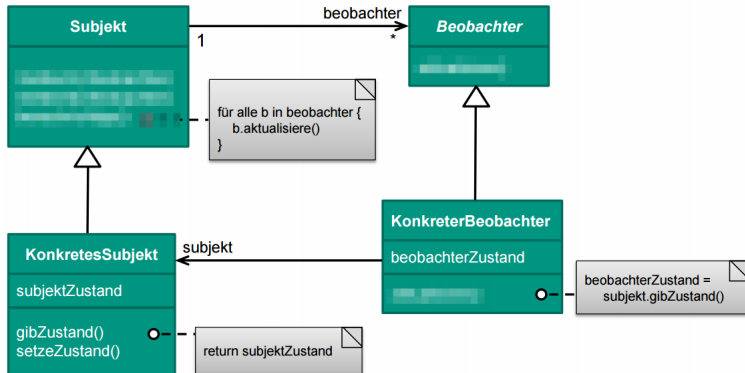
# Was bisher geschah..

- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



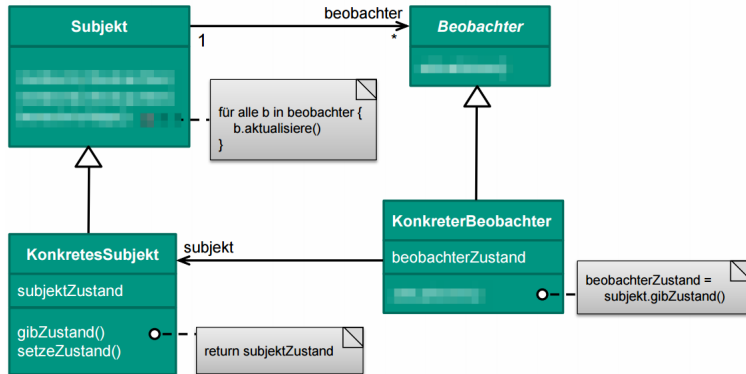
# Was bisher geschah..

- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



# Was bisher geschah..

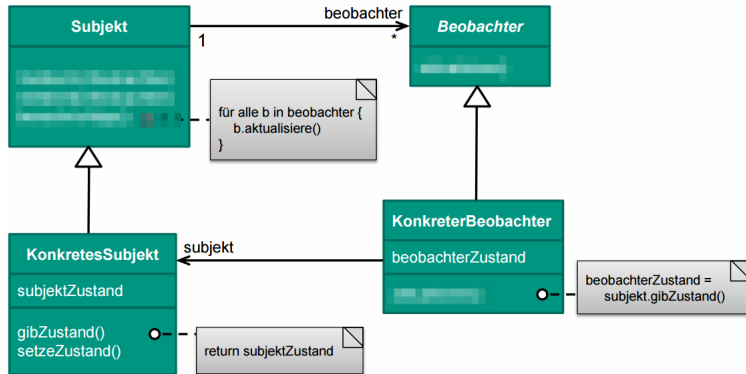
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



Ist wohl ein Beobachter :)

# Was bisher geschah..

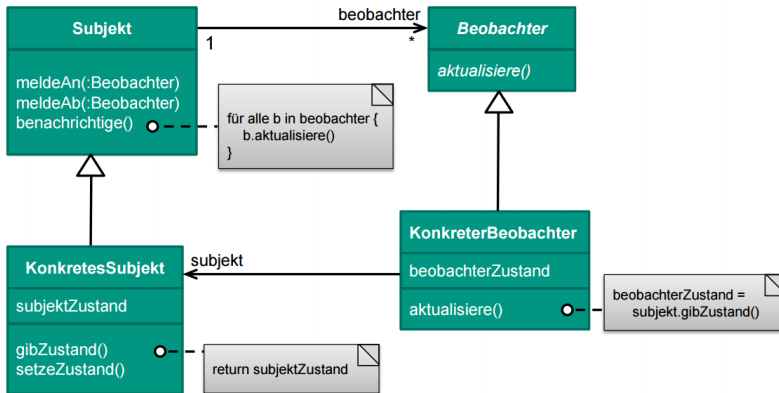
- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



Ist wohl ein Beobachter :) Klassen, Methoden?

# Was bisher geschah..

- haben uns erste Entkopplungsmuster angeschaut  
⇒ Beobachter, Iterator, Adapter



## ■ Entkopplungs-Muster

- Adapter fertig
- Beobachter fertig
- Iterator fertig
- **Stellvertreter**
- **Vermittler**
- (Brücke)

## ■ Varianten-Muster

## ■ Zustandshandhabungs-Muster

## ■ Steuerungs-Muster

## ■ Bequemlichkeits-Muster

## Problem

- wollen Zugriff auf ein Objekt kontrollieren, ohne seine Klasse zu ändern

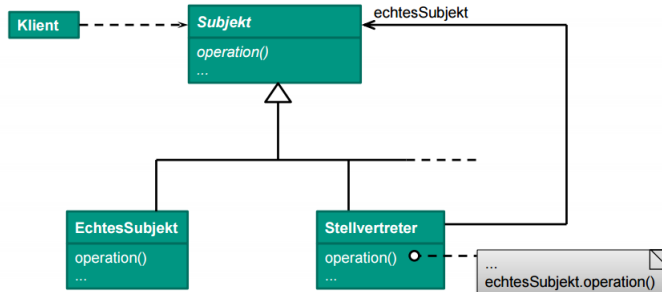
## Problem

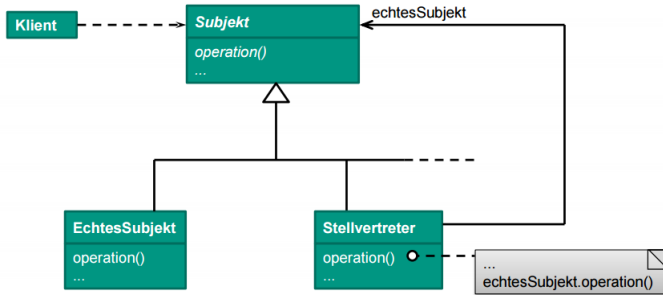
- wollen Zugriff auf ein Objekt kontrollieren, ohne seine Klasse zu ändern  
⇒ Stellvertreter macht Zugriffskontrolle



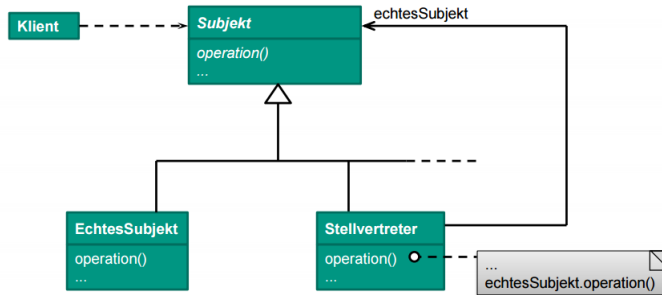
## Problem

- wollen Zugriff auf ein Objekt kontrollieren, ohne seine Klasse zu ändern  
⇒ Stellvertreter macht Zugriffskontrolle



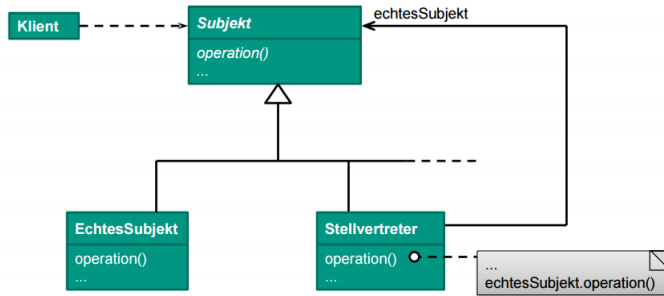


## Entkopplung?



## Entkopplung?

- Klient hat keinen direkten Zugriff auf das echte Subjekt



## Entkopplung?

- Klient hat keinen direkten Zugriff auf das echte Subjekt
- Stellvertreter hat Relation zu Oberklasse (!), echtes Subjekt austauschbar

## Problem

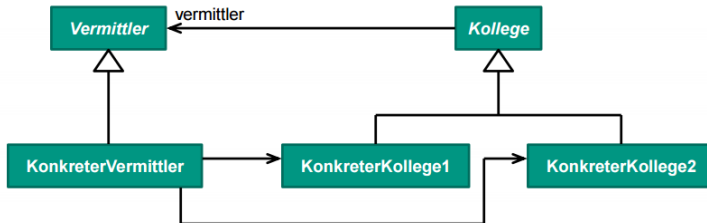
- mehrere voneinander abhängige Objekte

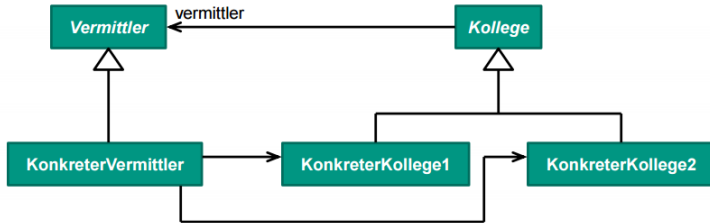
## Problem

- mehrere voneinander abhängige Objekte  
     $\Rightarrow$  Zustände der Objekte von anderen Zuständen abhängig

## Problem

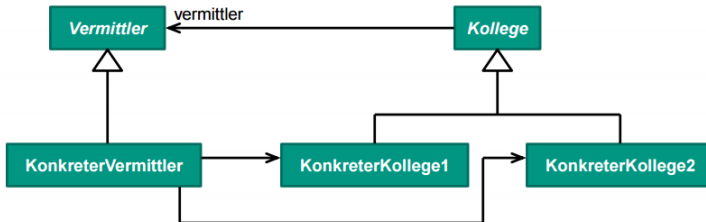
- mehrere voneinander abhängige Objekte  
⇒ Zustände der Objekte von anderen Zuständen abhängig





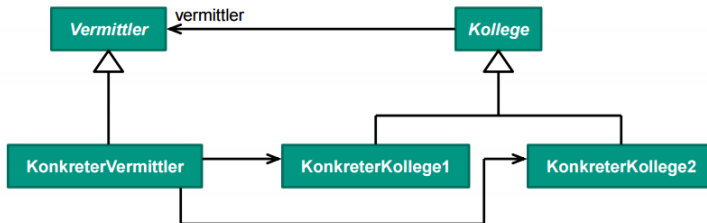
## Entkopplung?





## Entkopplung?

- Kollegen kennen sich nicht direkt



## Entkopplung?

- Kollegen kennen sich nicht direkt  
⇒ Hinzufügen eines Kollegen erfordert keine Änderung der alten Kollegen

- Entkopplungs-Muster **fertig**
- **Varianten-Muster**
  - Abstrakte Fabrik
  - Besucher
  - Schablonenmethode
  - Fabrikmethode
  - Kompositum
  - Strategie **fertig**
  - Dekorierer
- Zustandshandhabungs-Muster
- Steuerungs-Muster
- Bequemlichkeits-Muster

# Jetzt: Gruppenarbeit

- 1 ich teile pro Reihe ein Aufgabenblatt aus
- 2 ihr habt Zeit zum Bearbeiten
- 3 ihr stellt den anderen eure Lösung vor

## Aufgabe 1:



## Aufgabe 1:



## Aufgabe 2:



## Aufgabe 3:



## Aufgabe 3:



## Aufgabe 4:

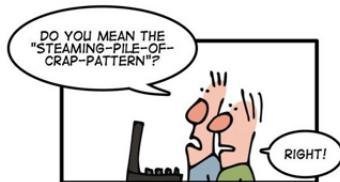




## Abgabe

- Deadline am 5.7. um 12:00

# Bis dann! (dann := 10.07.17)



THE HYPE IS LONG GONE BUT  
DESIGN PATTERNS ARE STILL USEFUL