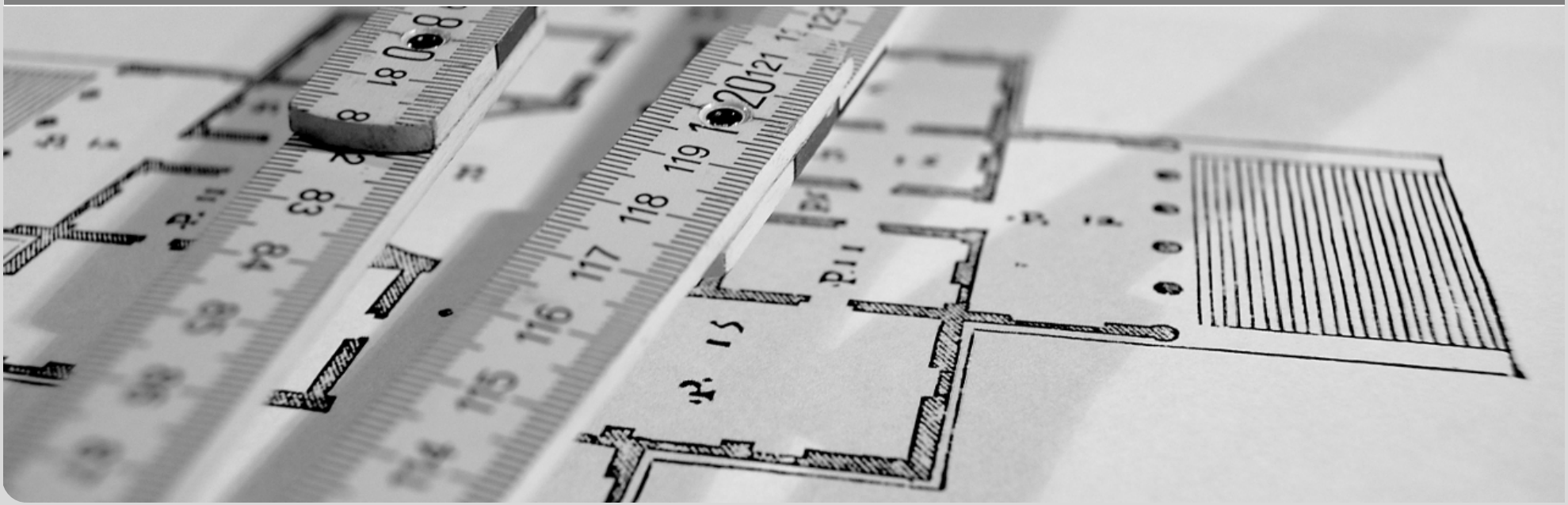
Karlsruher Institut für Technologie

Softwaretechnik 1 - 4. Tutorium

Tutorium 03

Felix Bachmann | 26.06.2017

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



Allgemeines

Ansage der Übungsleiter

- ab jetzt keine Abgabe per Mail mehr!
  $\Longrightarrow$ auch nicht in Ausnahmefällen

4. Übungsblatt Statistik

Häufige Fehler

Allgemein

Häufige Fehler

Aufgabe 1 (Zustandsdiagramm - LEZ): ∅

Häufige Fehler

Aufgabe 2 (Abbottsche Methode): ∅

Aufgabe 3 (iMage-GUI): ∅

Häufige Fehler

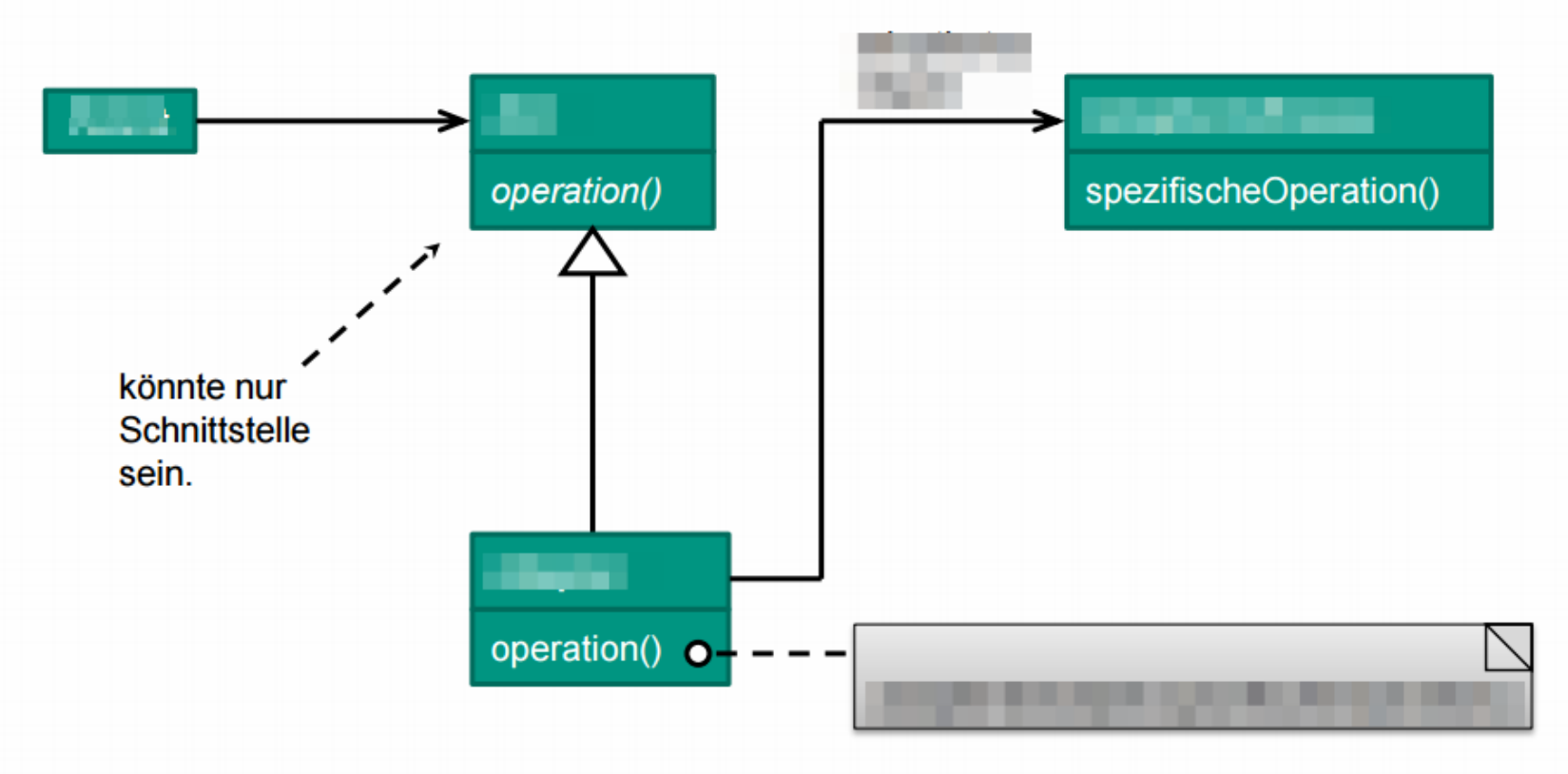
Aufgabe 4 (Geheimnisprinzip): ∅

Was bisher geschah..

- haben uns erste Entkopplungmuster angeschaut

Was bisher geschah..

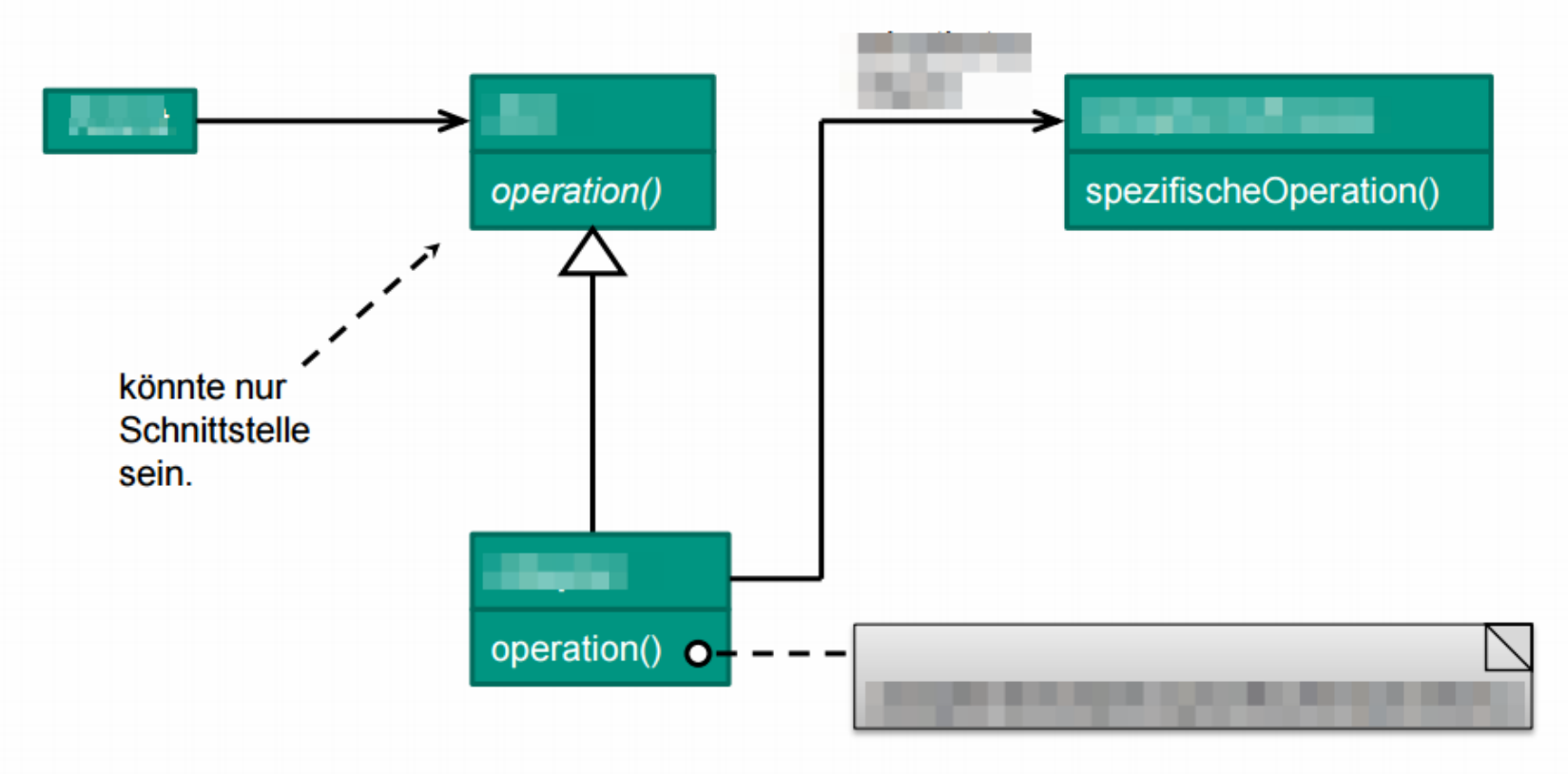
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Welches Entwurfsmuster?

Was bisher geschah..

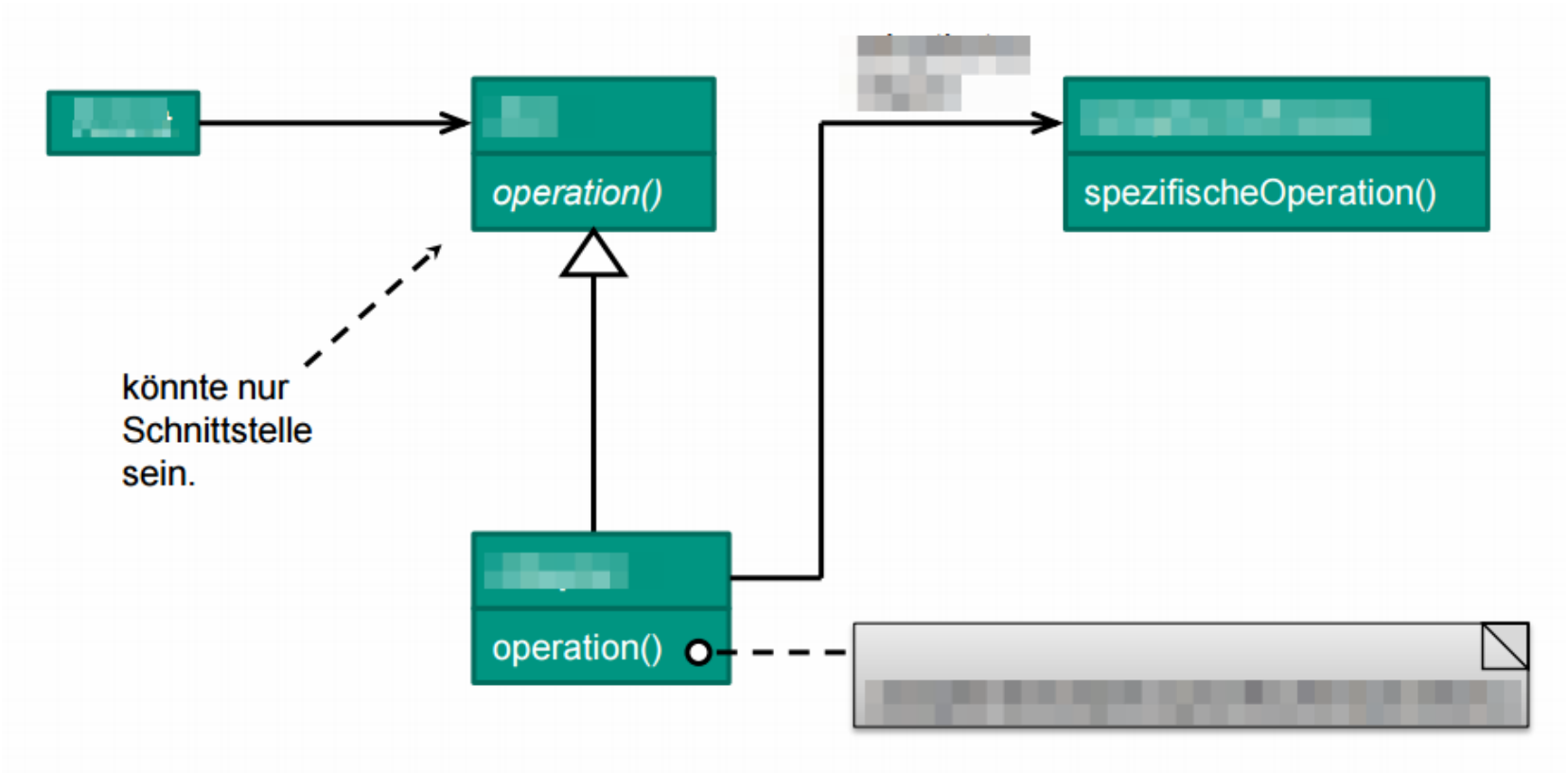
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Welches Entwurfsmuster? (Objekt-)Adapter

Was bisher geschah..

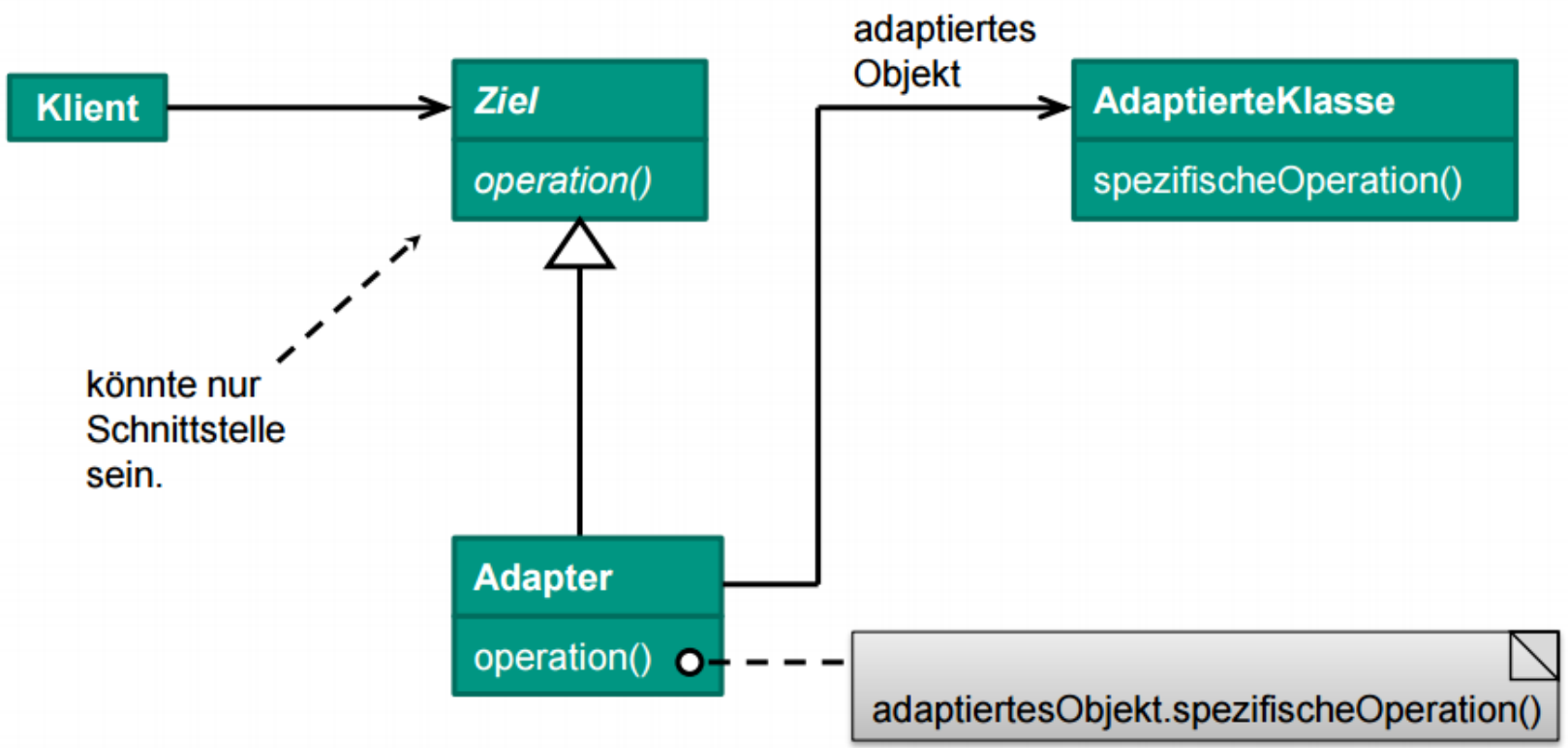
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Welche Klassen?

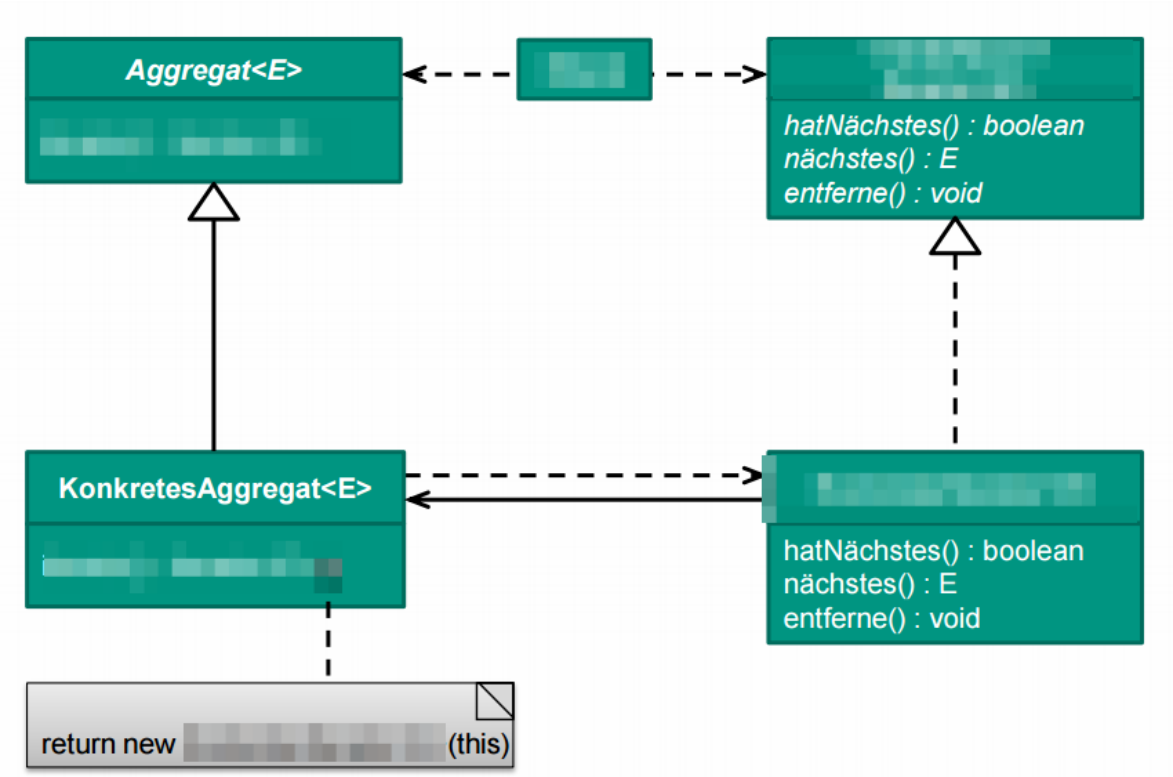
Was bisher geschah..

- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Was bisher geschah..

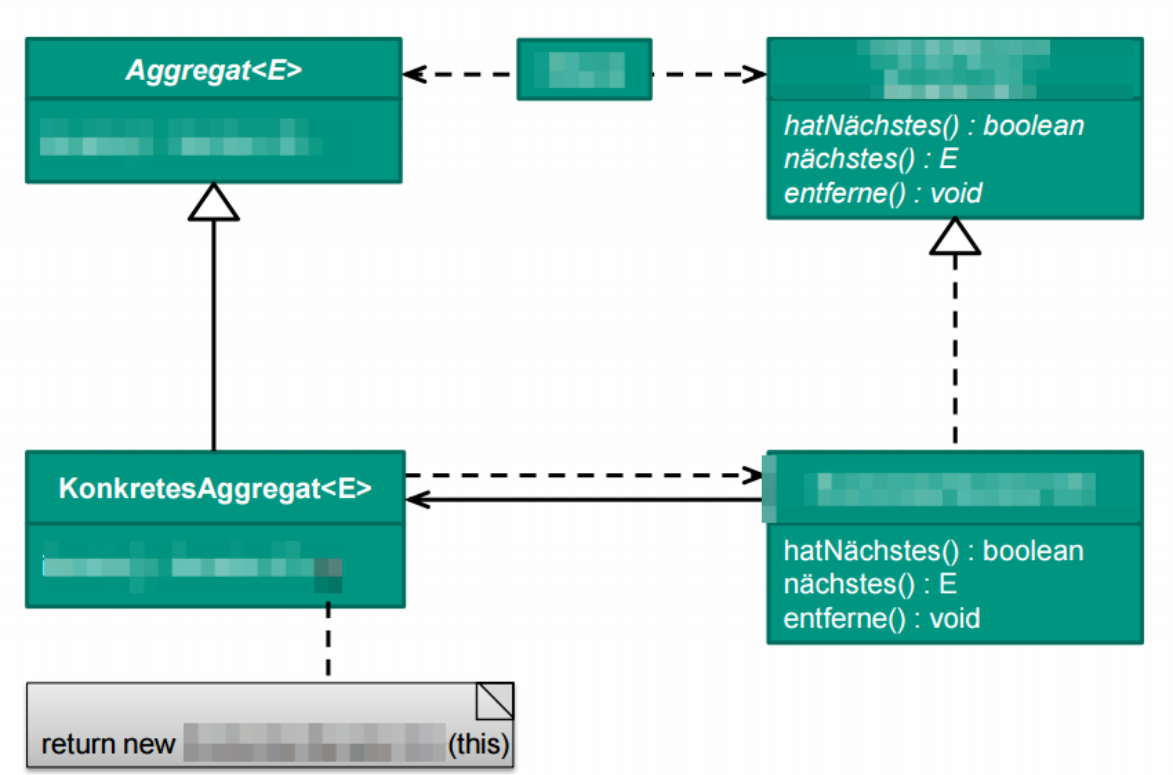
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Welches Entwurfsmuster?

Was bisher geschah..

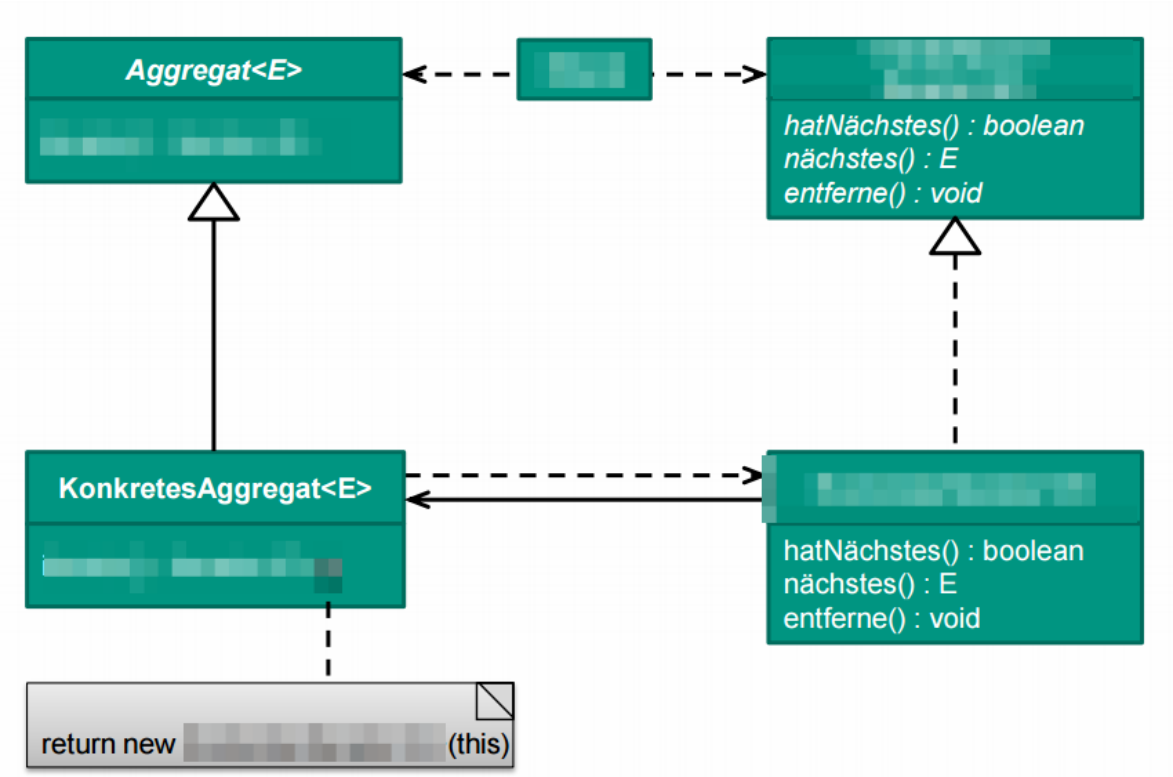
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Welches Entwurfsmuster? Iterator

Was bisher geschah..

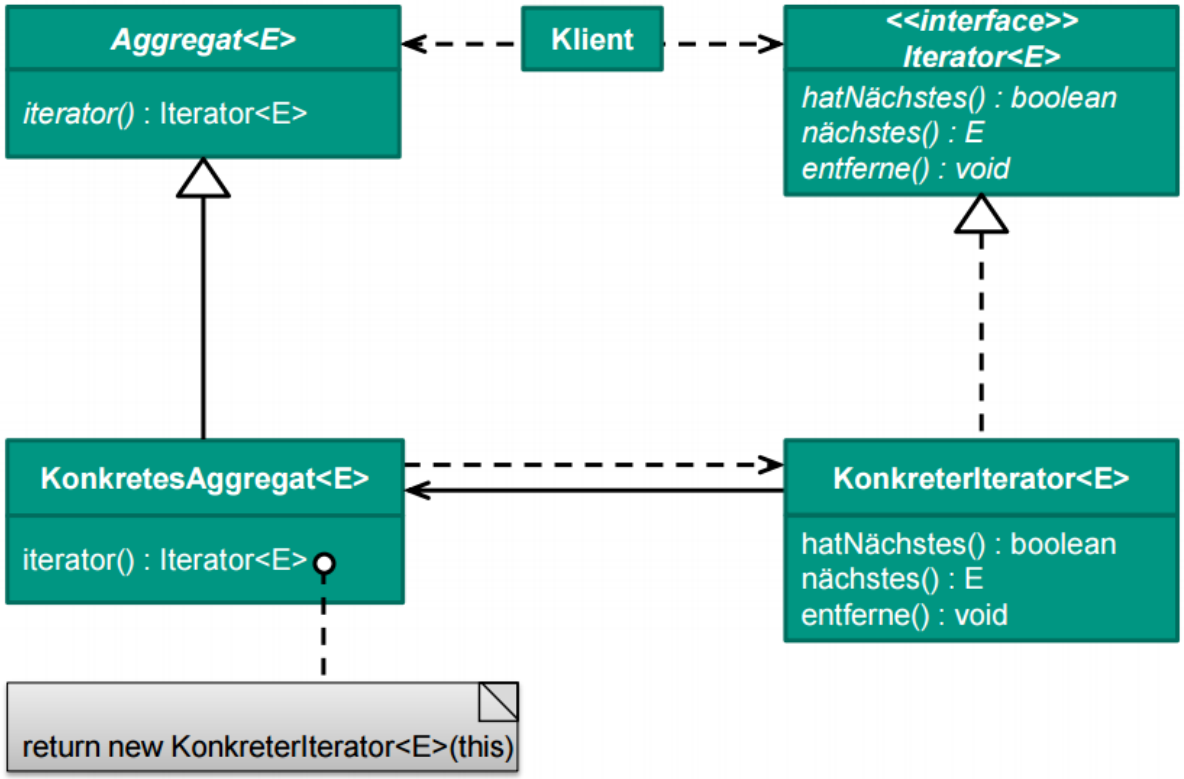
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Welche Klassen und Methoden?

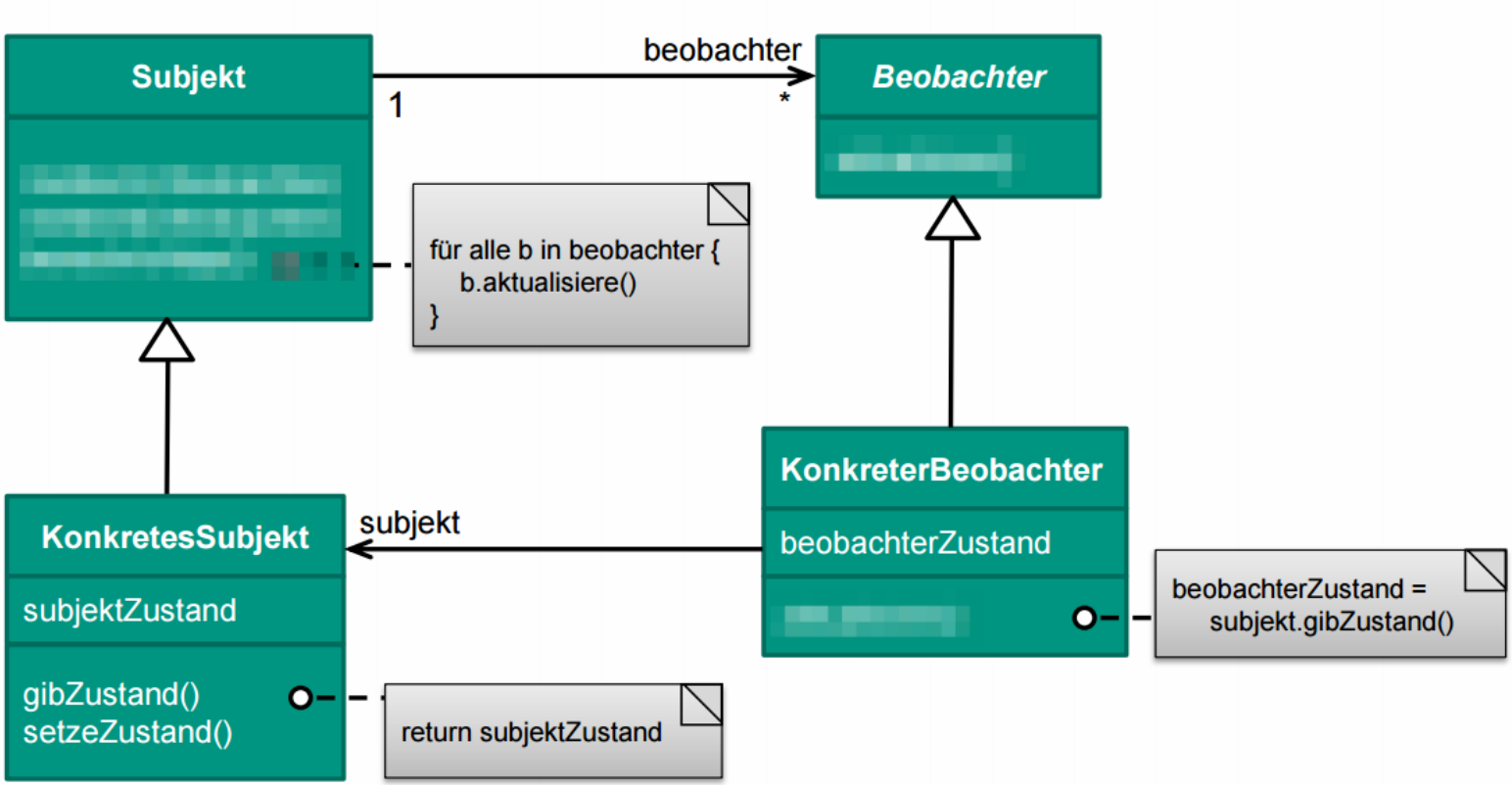
Was bisher geschah..

- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



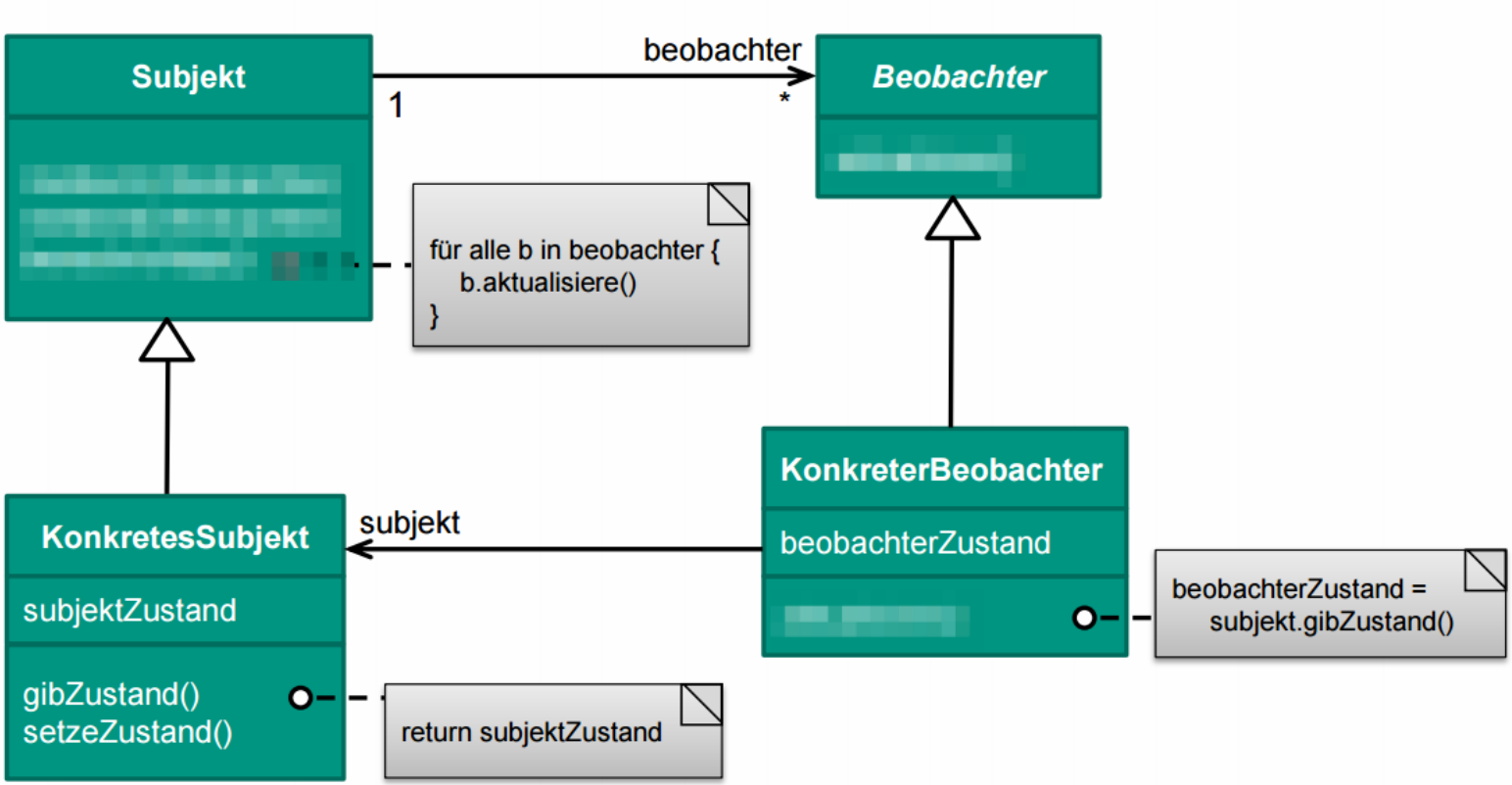
Was bisher geschah..

- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Was bisher geschah..

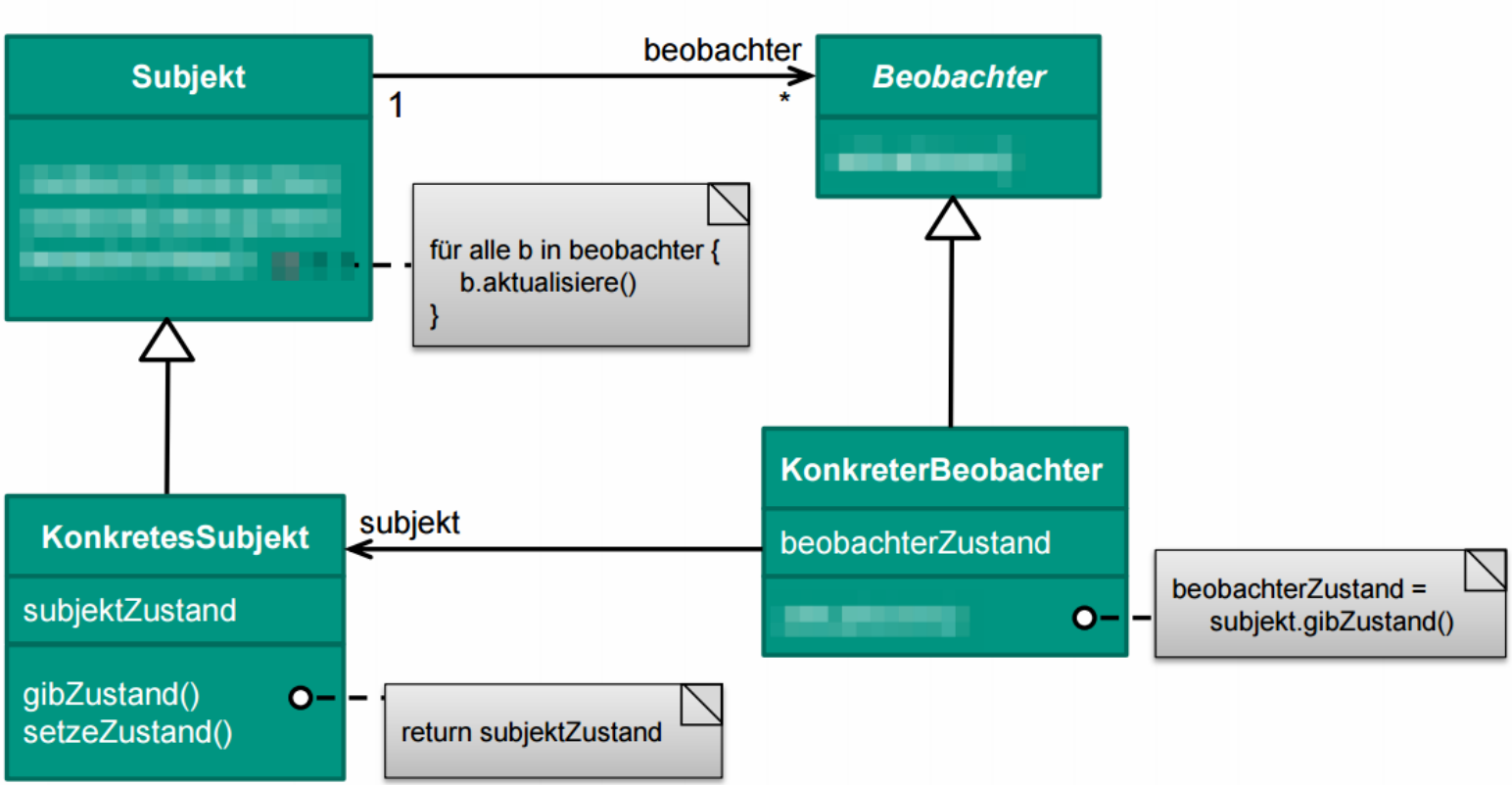
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Ist wohl ein Beobachter :)

Was bisher geschah..

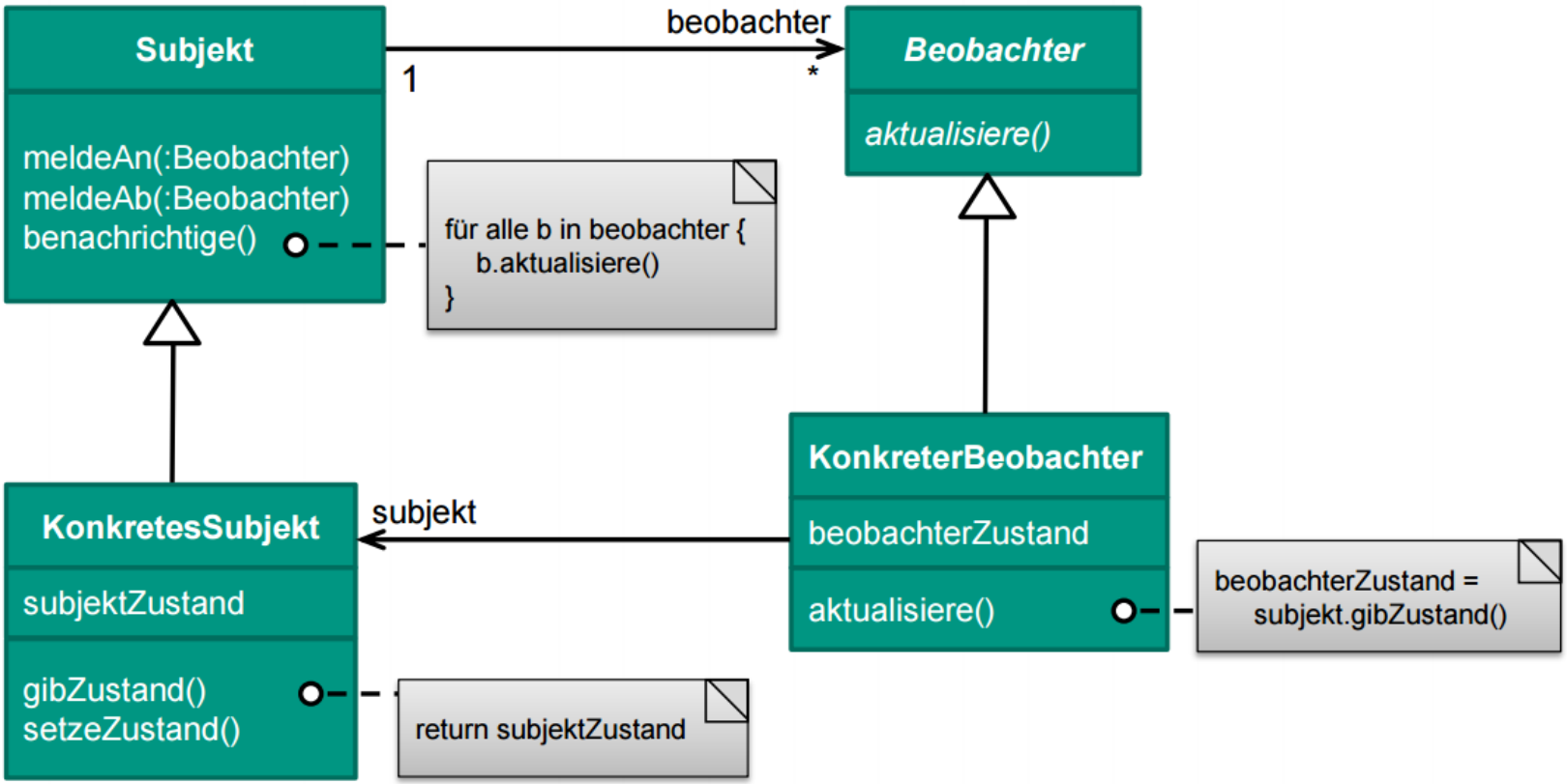
- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Ist wohl ein Beobachter :) Klassen, Methoden?

Was bisher geschah..

- haben uns erste Entkopplungmuster angeschaut
  $\Longrightarrow$ Beobachter, Iterator, Adapter



Kategorien der Entwurfsmuster

- **Entkopplungs-Muster**
  - Adapter fertig
  - Beobachter fertig
  - Iterator fertig
  - **Stellvertreter**
  - **Vermittler**
  - (Brücke)
- Varianten-Muster
- Zustandshandhabungs-Muster
- Steuerungs-Muster
- Bequemlichkeits-Muster

Stellvertreter

Problem

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

Stellvertreter

Problem

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

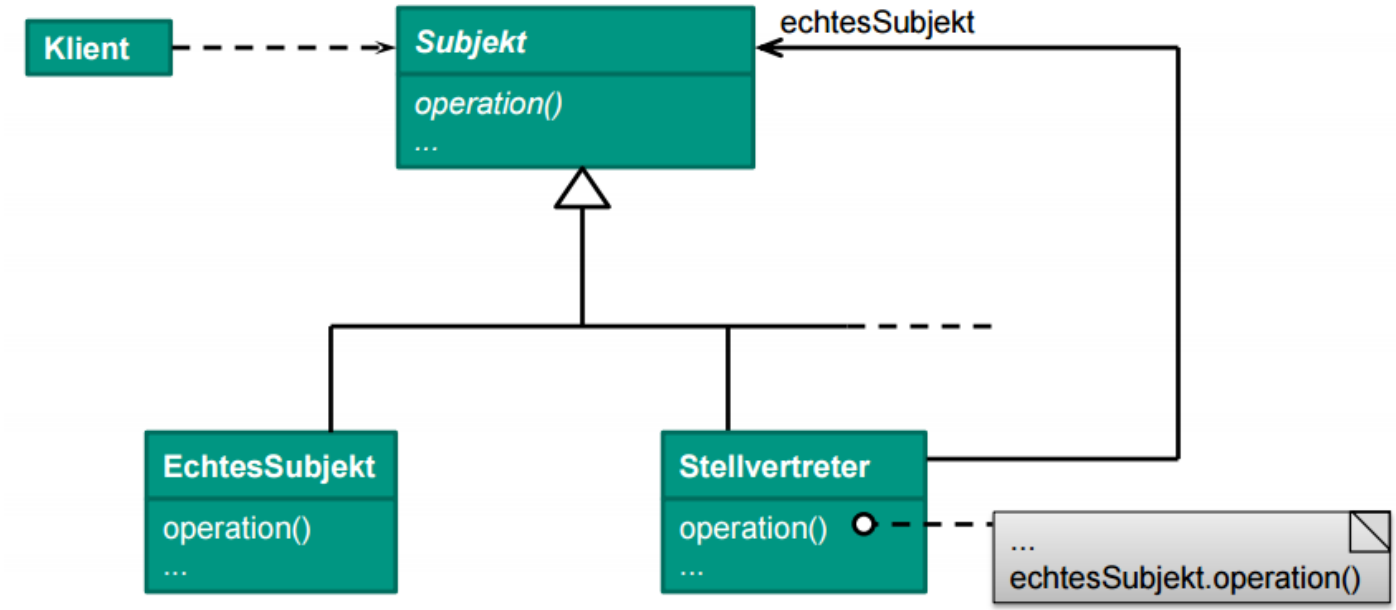
  $\Longrightarrow$ Stellvertreter macht Zugriffskontrolle

Stellvertreter

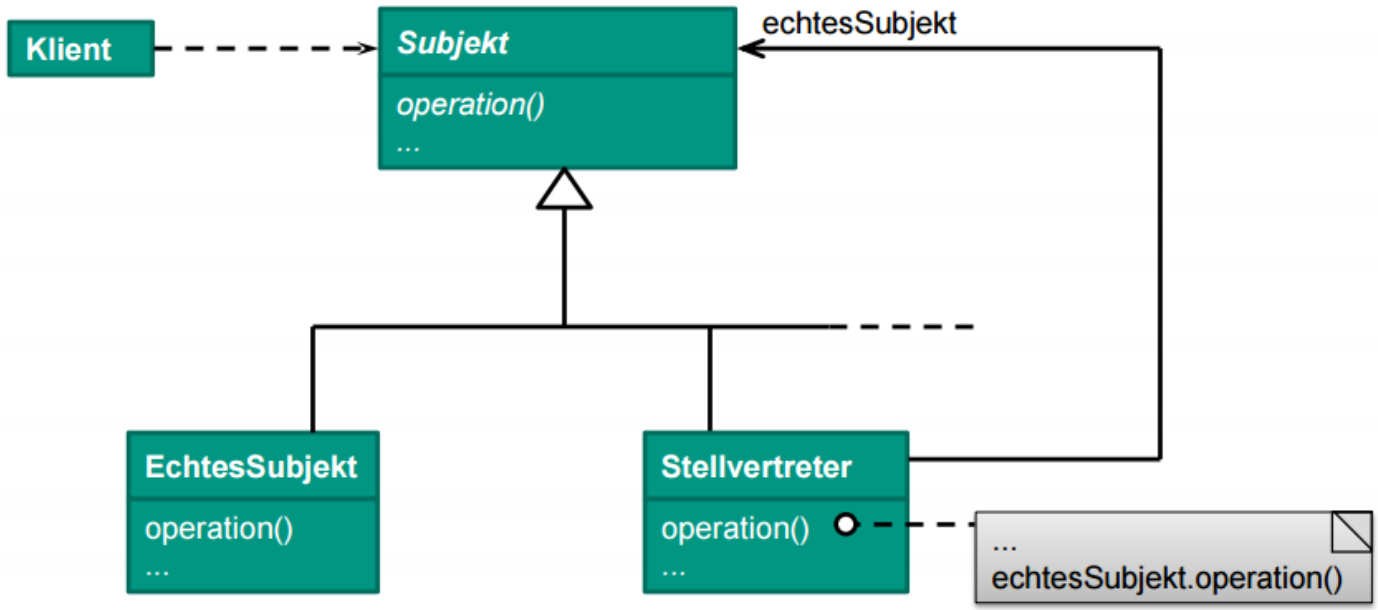
Problem

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

  $\Longrightarrow$ Stellvertreter macht Zugriffskontrolle

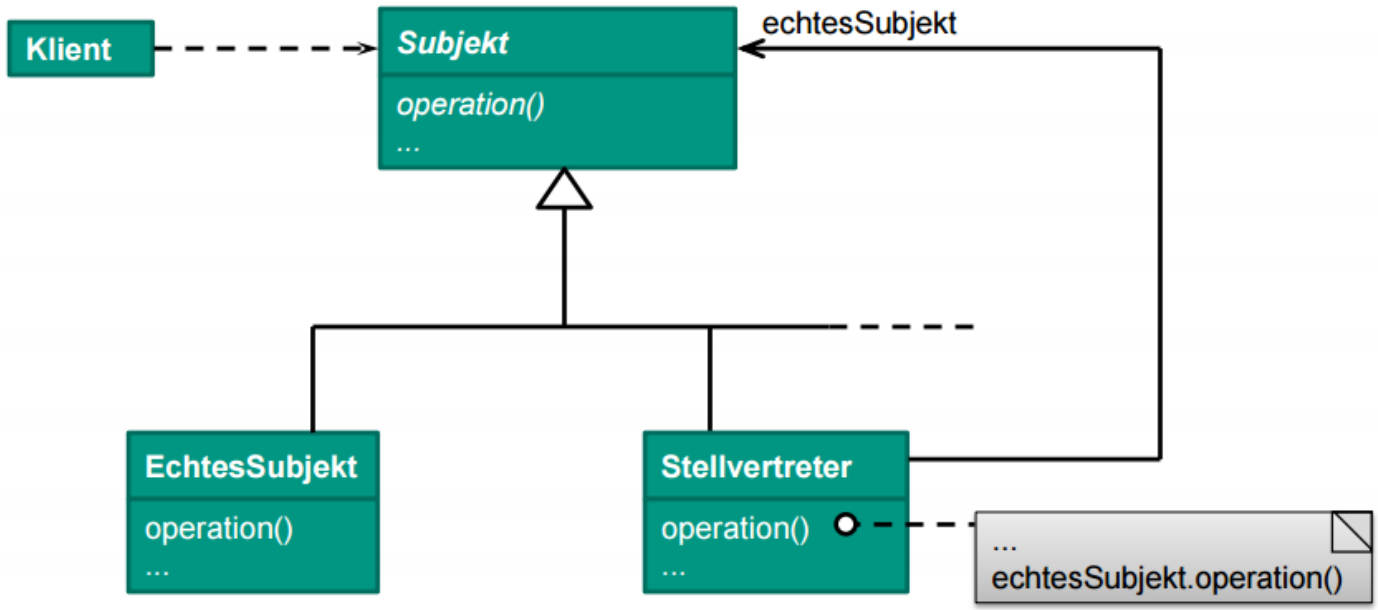


Stellvertreter



Entkopplung?

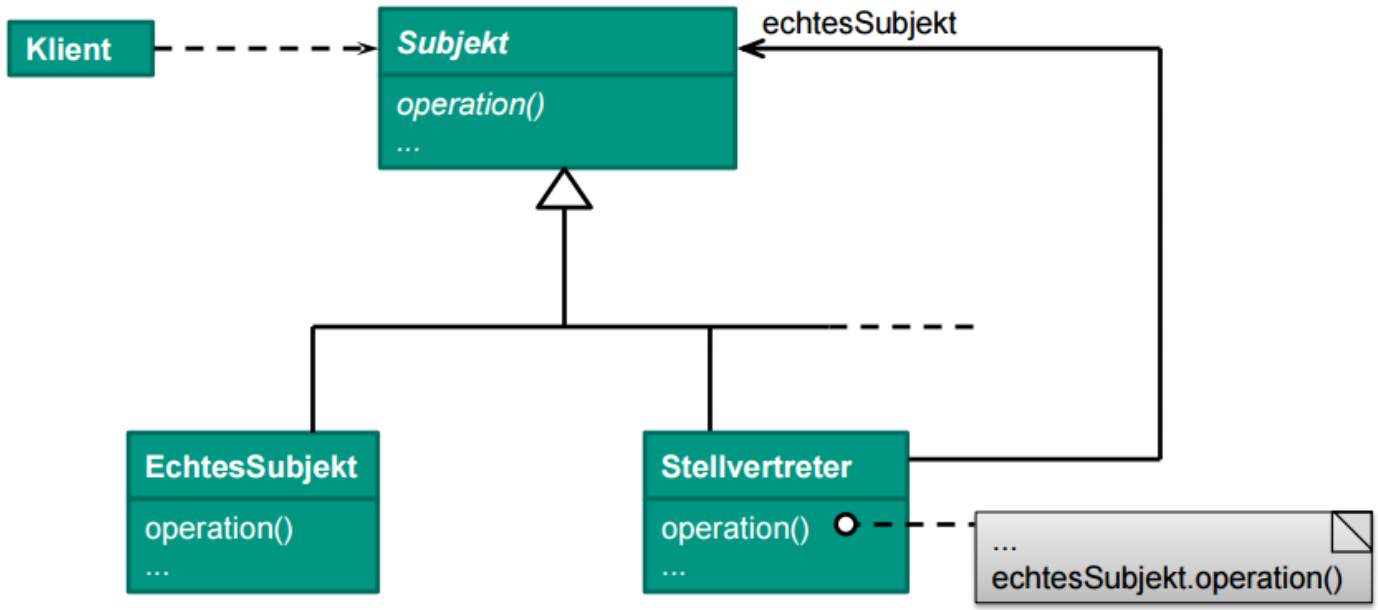
Stellvertreter



Entkopplung?

- Klient hat keinen direkten Zugriff auf das echte Subjekt

Stellvertreter



Entkopplung?

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

Vermittler

Problem

- mehrere voneinander abhängige Objekte

Vermittler

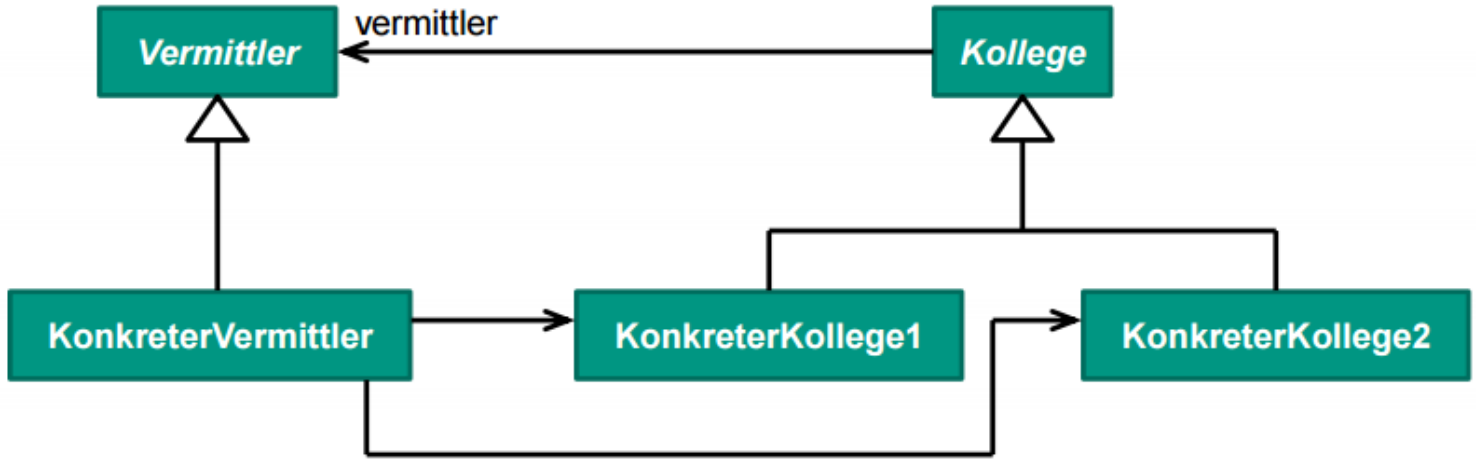
Problem

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

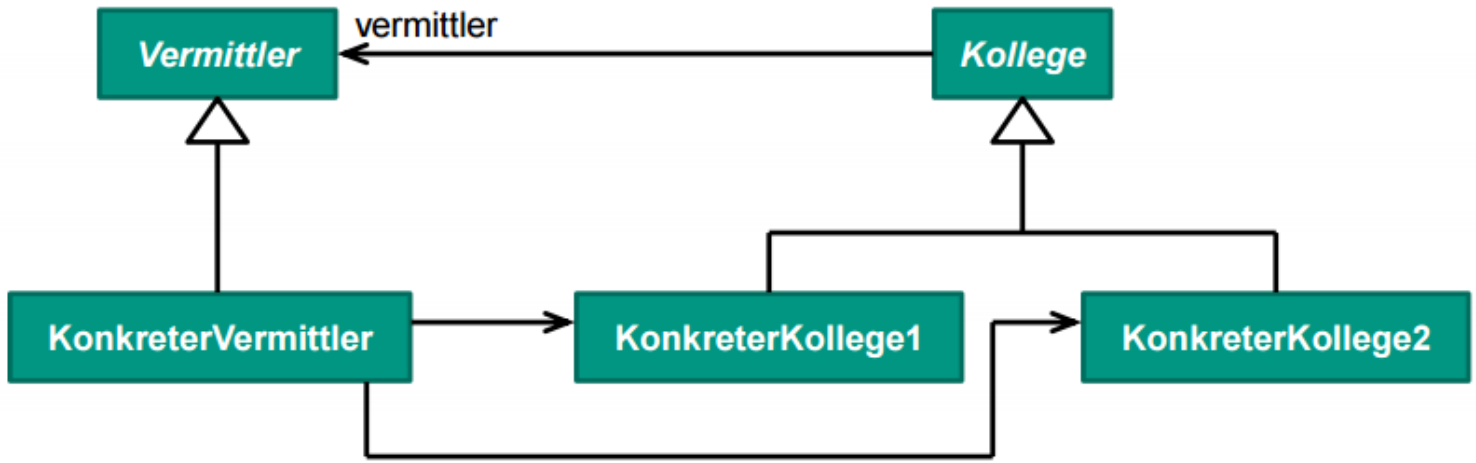
Vermittler

Problem

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

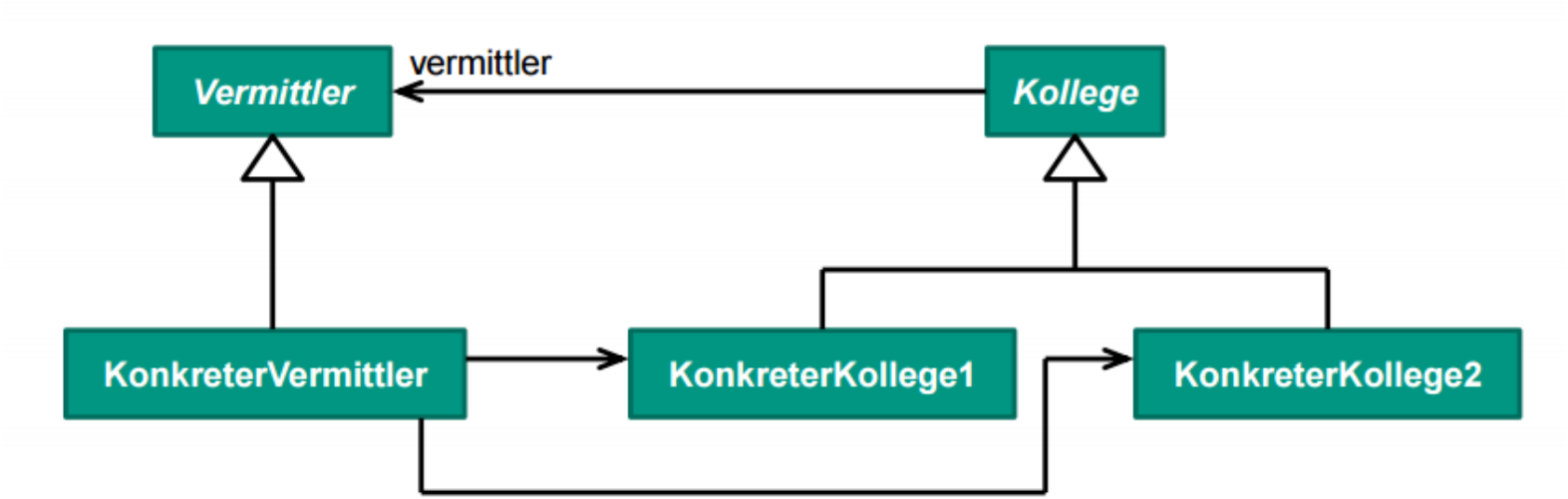


Vermittler



Entkopplung?

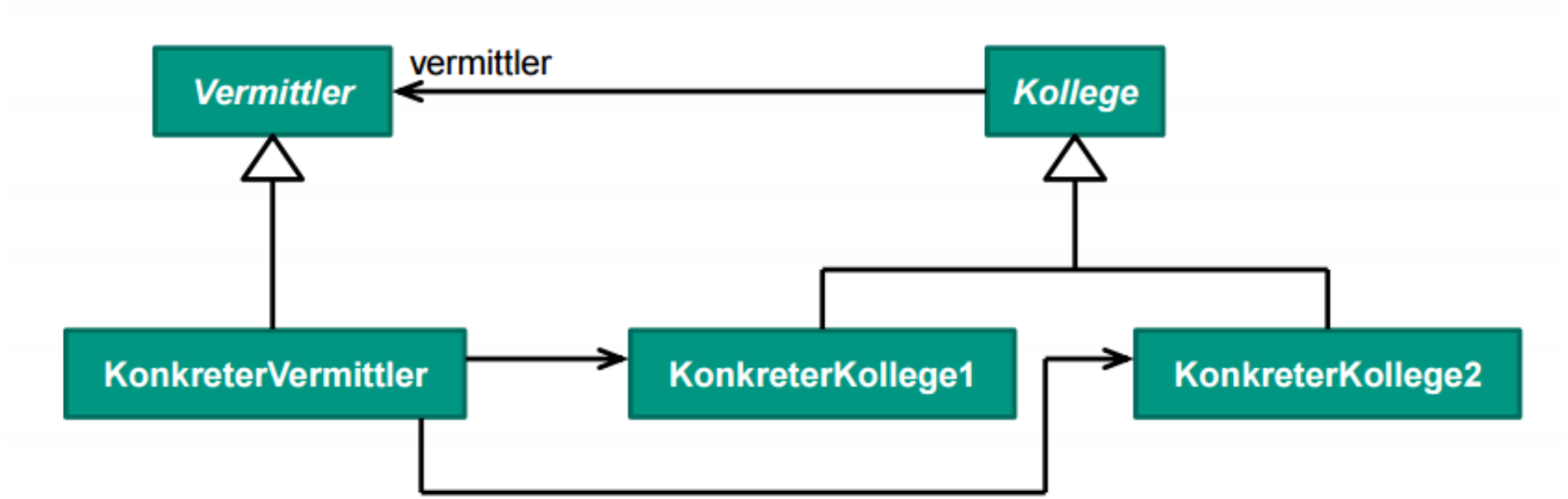
Vermittler



Entkopplung?

- Kollegen kennen sich nicht direkt

Vermittler



Entkopplung?

- Kollegen kennen sich nicht direkt
  $\Longrightarrow$ Hinzufügen eines Kollegen erfordert keine Änderung der alten Kollegen

Kategorien der Entwurfsmuster

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

Varianten-Muster

Übergeordnetes Ziel

- übergeordnetes Ziel: Gemeinsamkeiten herausziehen und an einer Stelle beschreiben

Varianten-Muster

Übergeordnetes Ziel

- übergeordnetes Ziel: Gemeinsamkeiten herausziehen und an einer Stelle beschreiben
  $\Longrightarrow$ keine Wiederholung desselben Codes

Varianten-Muster

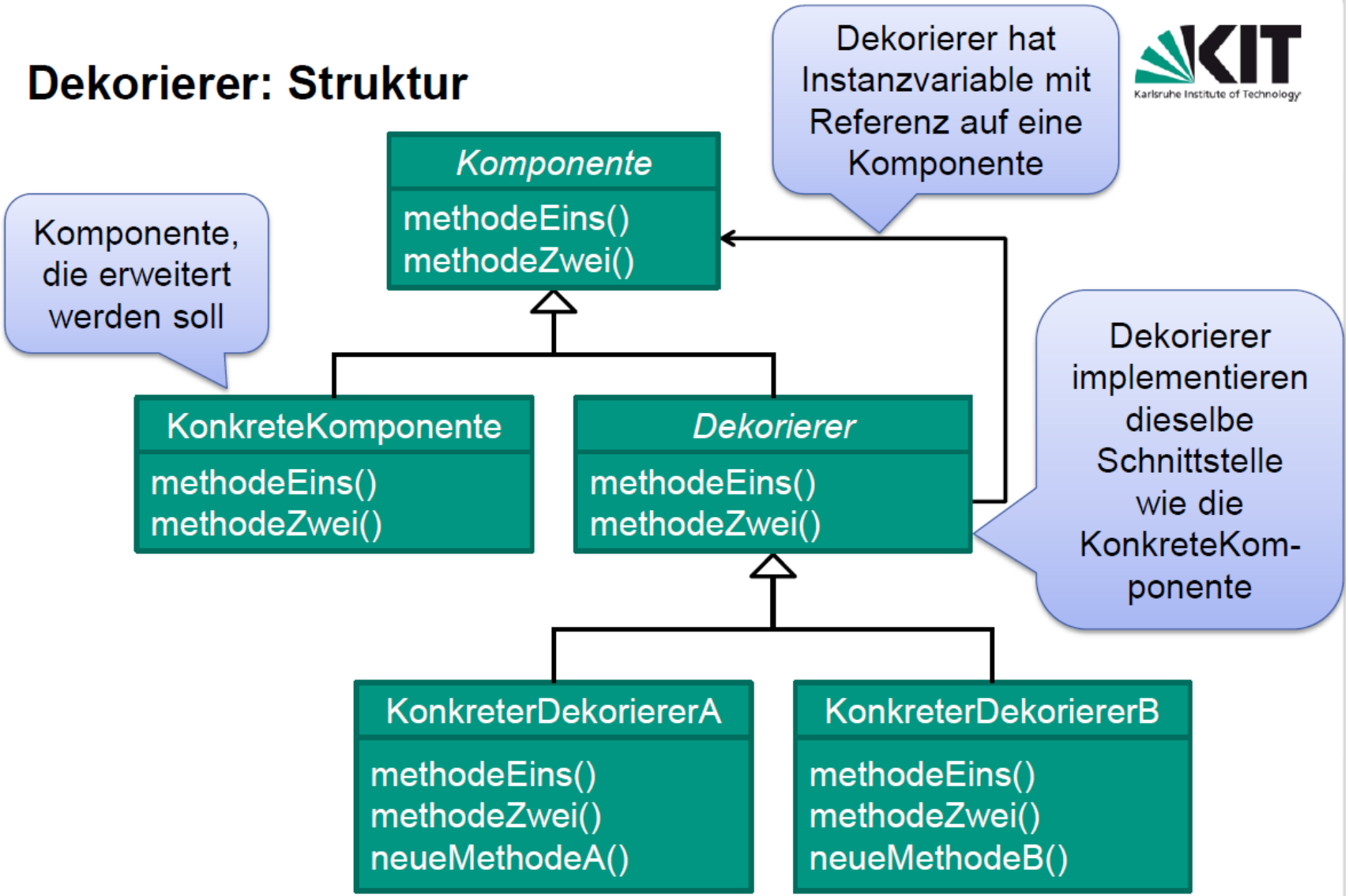
Übergeordnetes Ziel

- übergeordnetes Ziel: Gemeinsamkeiten herausziehen und an einer Stelle beschreiben
  $\Longrightarrow$ keine Wiederholung desselben Codes
  $\Longrightarrow$ bessere Wartbarkeit/Erweiterbarkeit

Jetzt: Gruppenarbeit

1. ihr kriegt pro Reihe eine Aufgabe
2. ihr habt Zeit zum Bearbeiten
3. ihr stellt den anderen eure Lösung vor

Vorstellung Dekorierer



MuLö Dekorierer

Wo Gemeinsamkeiten?

Die beiden Methoden methodeEins() und methodeZwei().

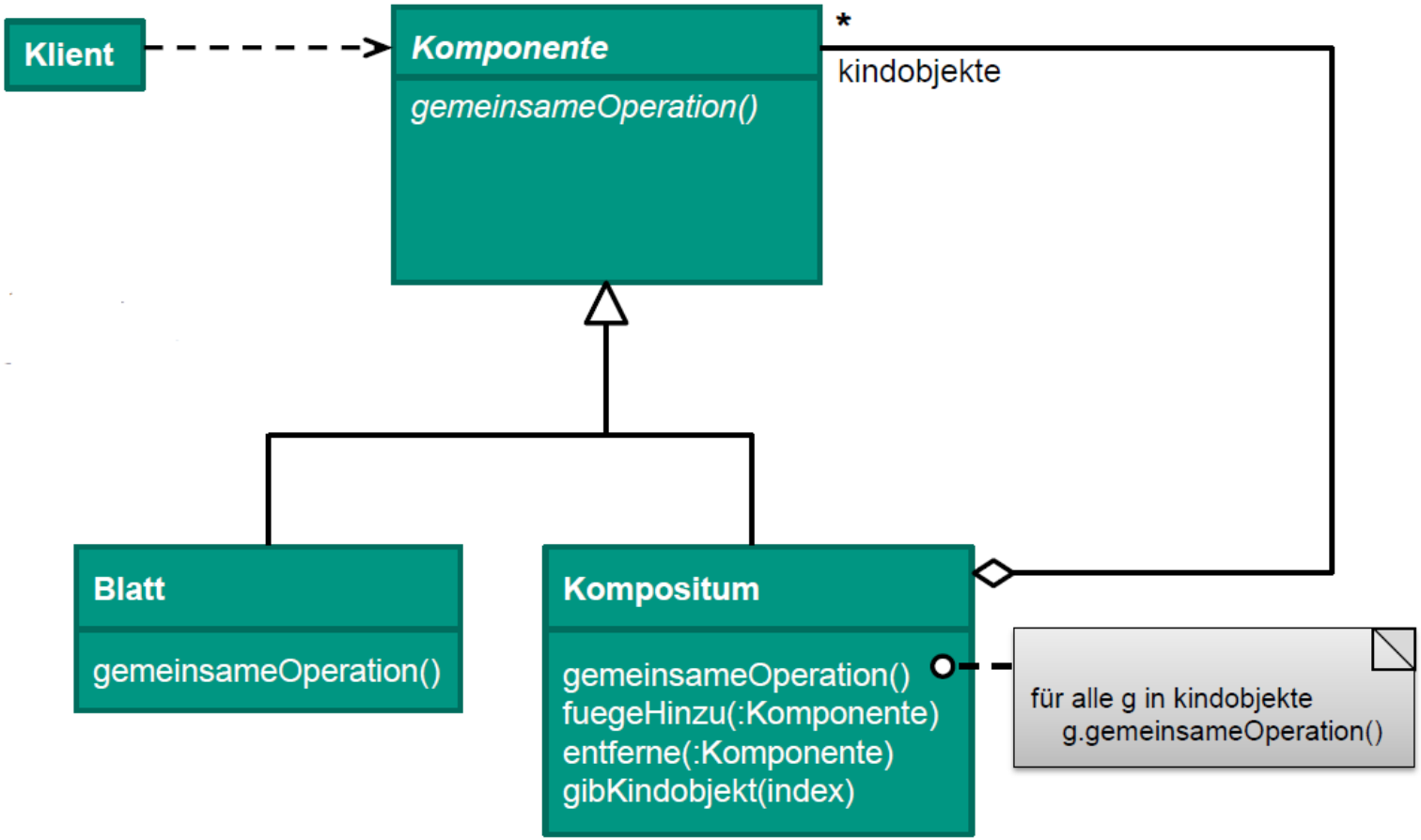
Wo Variation?

In den KonkretenDekorierern bzw. ihren Methoden. Hier: neueMethodeA(), neueMethodeB().

Wozu Instanzvariable?

Weiterleitung von Aufrufen der methodeEins() und methodeZwei() an die KonkreteKompenente.

Vorstellung Kompositum



MuLö Kompositum

Wo Gemeinsamkeiten?

gemeinsameOperation().

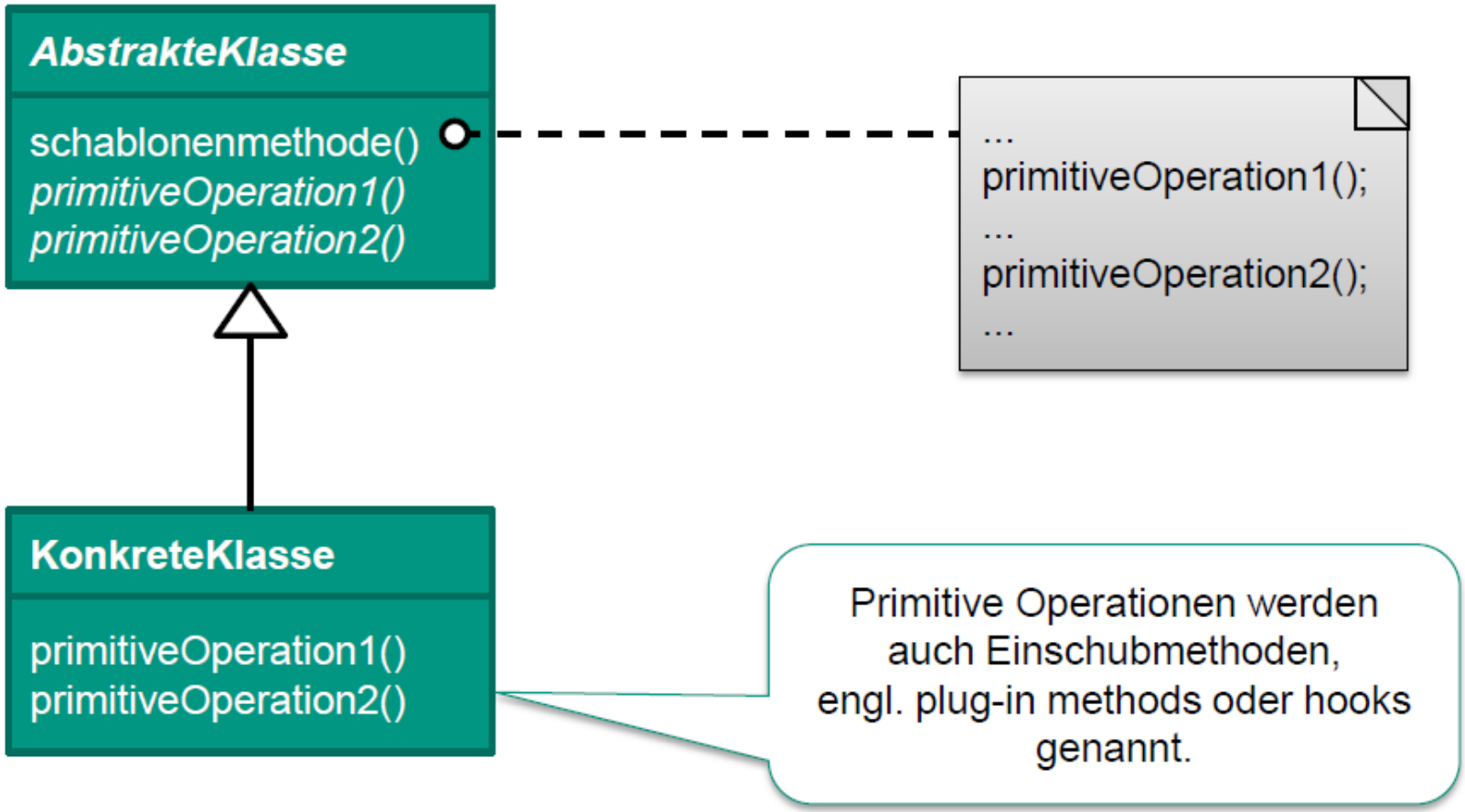
Wo Variation?

In Blatt/Kompositum-Klassen mit verschiedenen zusätzlichen Operationen.

Zusammengesetzt vs. nicht-zusammengesetzt

Kompositum = zusammengesetzt, Blatt = nicht-zusammengesetzt

Vorstellung Schablonenmethode



MuLö Schablonenmethode

Wo Gemeinsamkeiten?

Reihenfolge der Methodenaufrufe in der Schablonenmethode.

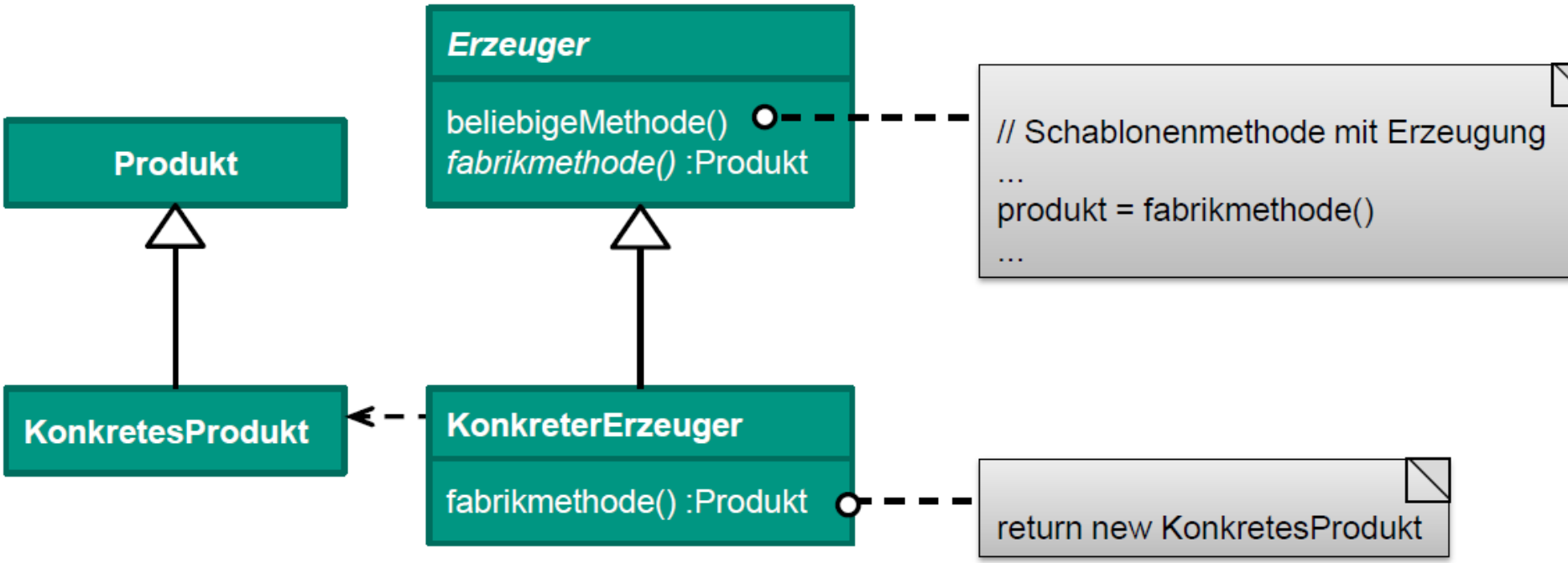
Wo Variation?

In den Einschubmethoden. (hier: primitiveOperation1() und primitiveOpoeration2())

Schablonenmethode vs. Einschubmethode

Einschubmethode ist eine der Methoden, die von der Schablonenmethode aufgerufen wird und deren Implementierung in den Unterklassen stattfindet.

Vorstellung Fabrikmethode



MuLö Fabrikmethode

Wo Gemeinsamkeiten?

Reihenfolge der Methodenaufrufe in der beliebigenMethode().

Wo Variation?

In der Fabrikmethode.

Klasse des Objekts, Oberklasse, Unterklasse

Klasse des Objekts = KonkretesProdukt, Oberklasse = Produkt, Unterklasse = KonkreterErzeuger

Unterschied zu Schablonenmethode?

Fabrikmethode benutzen, wenn ein Objekt erzeugt wird. Fabrikmethode ist Einschubmethode des Musters “Schablonenmethode“.

Wahr/falsch

Fabrikmethode ist eine Einschubmethode, keine Schablonenmethode.

Kategorien der Entwurfsmuster

- Entkopplungs-Muster fertig
- Varianten-Muster fertig
- **Zustandshandhabungs-Muster**
  - (Einzelstück)
  - (Fliegengewicht)
  - **Memento**
  - (Prototyp)
  - (Zustand)
- Steuerungs-Muster
- Bequemlichkeits-Muster

Zustandshandhabungs-Muster

Übergeordnetes Ziel

- den Zustand eines Objektes beschreiben (wer hätt's gedacht? :D)

Zustandshandhabungs-Muster

Übergeordnetes Ziel

- den Zustand eines Objektes beschreiben (wer hätt's gedacht? :D)
- aber unabhängig von dem Zweck des Objekts!

Memento

Problem

- internen Zustand eines Objekts “externalisieren“, um z.B. Zurücksetzen möglich zu machen

Memento

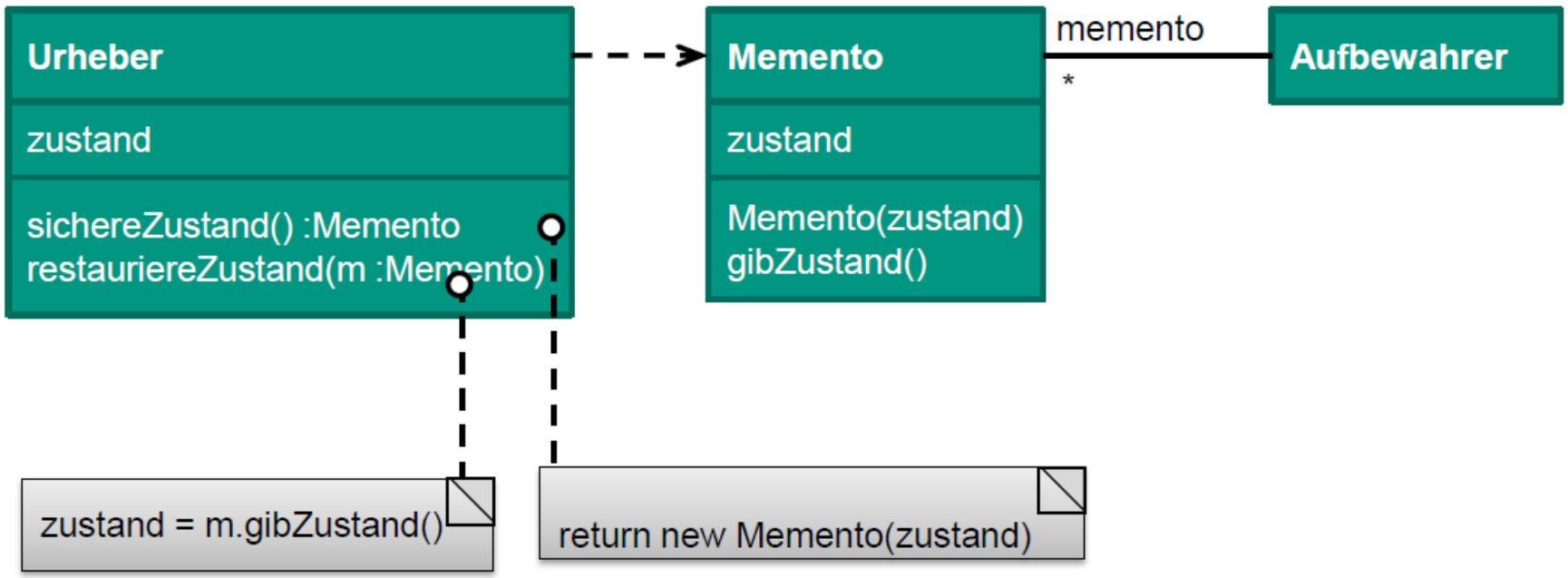
Problem

- internen Zustand eines Objekts “externalisieren“, um z.B. Zurücksetzen möglich zu machen
- ohne Kapselung zu verletzten!

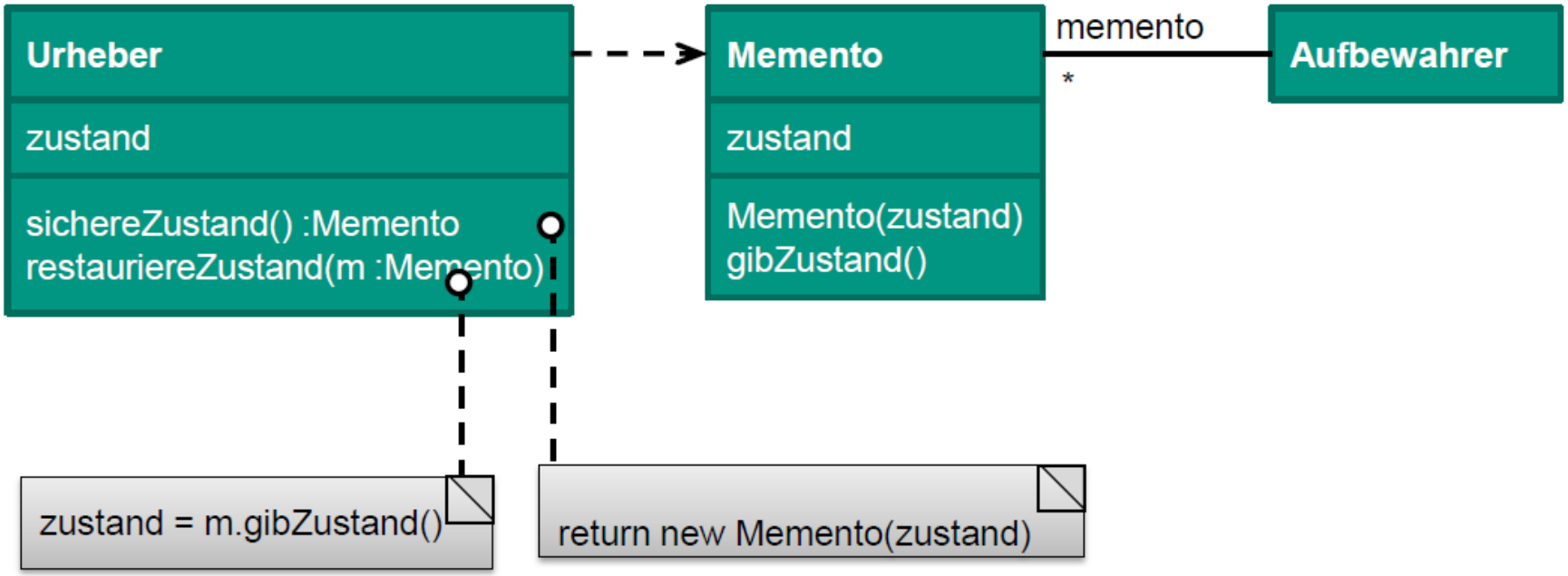
Memento

Problem

- internen Zustand eines Objekts “externalisieren“, um z.B. Zurücksetzen möglich zu machen
- ohne Kapselung zu verletzten!

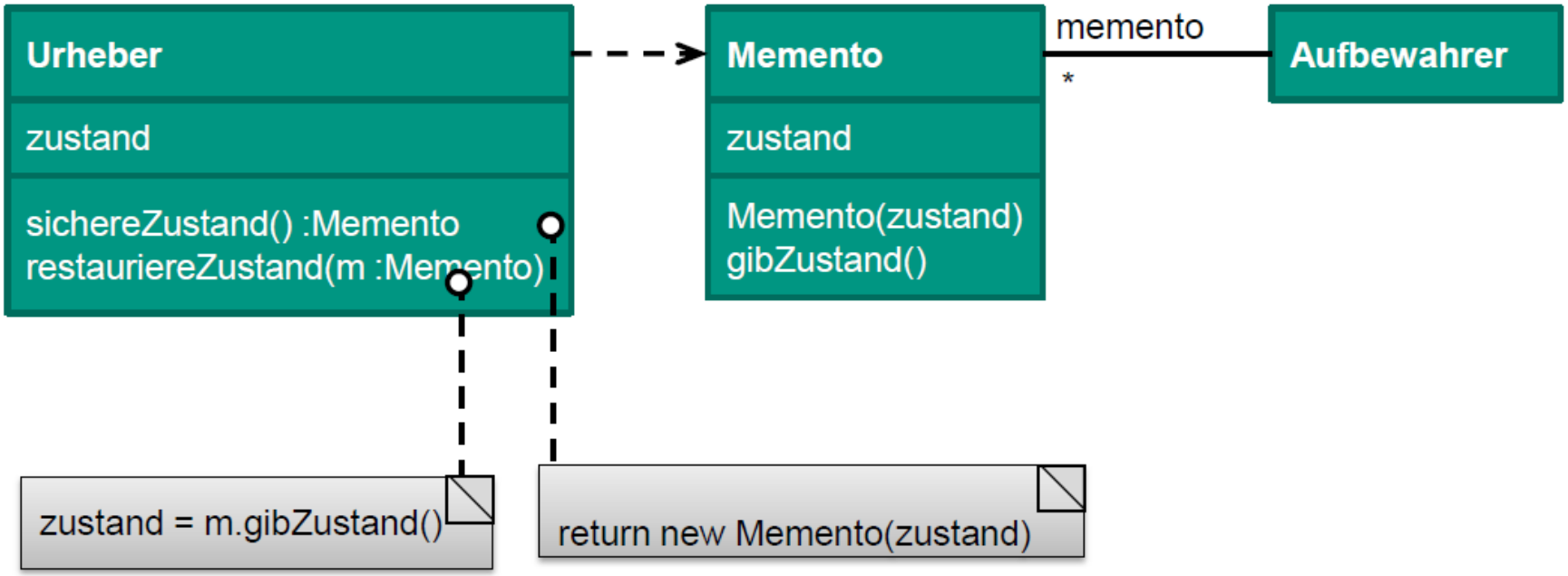


Memento



Problem gelöst?

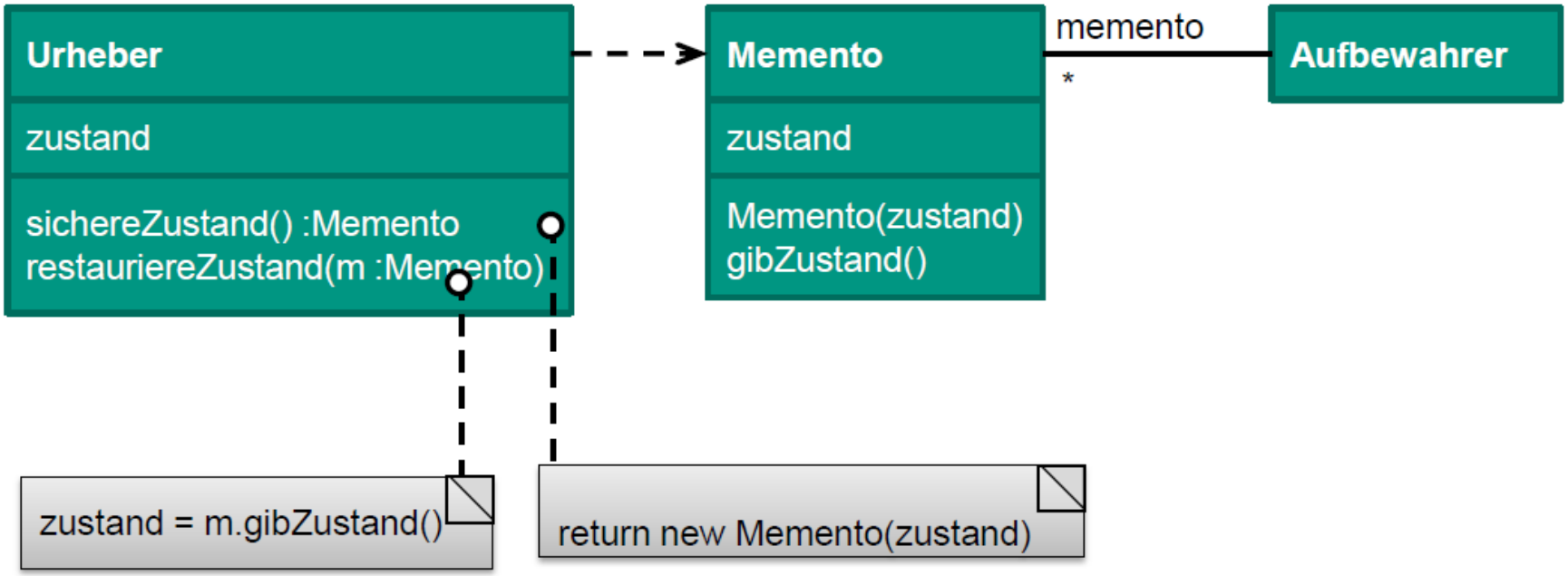
Memento



Problem gelöst?

- Ja

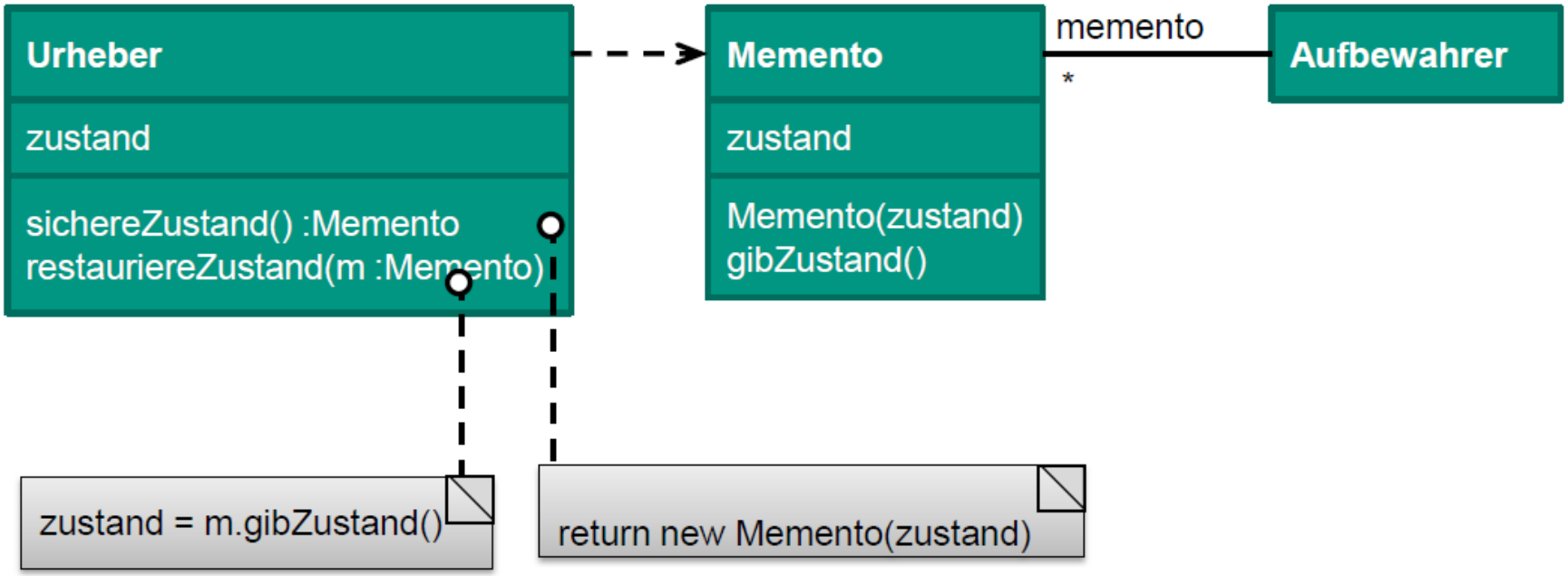
Memento



Problem gelöst?

- Ja
  - Zustand durch Memento externalisiert

Memento



Problem gelöst?

- Ja
  - Zustand durch Memento externalisiert
  - Kapselung nicht verletzt (Nutzer ruft nur sichereZustand() auf und kriegt neuen Memento)

Tipps - 5. Übungsblatt

Aufgabe 1: Manager-Deutsch und Architekturstile

- Architekturstile nochmal anschauen

Tipps - 5. Übungsblatt

Aufgabe 1: Manager-Deutsch und Architekturstile

- Architekturstile nochmal anschauen

Aufgabe 2: Iterator für Plug-Ins

- Iterator-Muster selbst benutzen

Tipps - 5. Übungsblatt

Aufgabe 3: Geometrify mit Entwurfsmustern

- überlegen, welches Entwurfsmuster **warum** Sinn macht

Tipps - 5. Übungsblatt

Aufgabe 3: Geometrify mit Entwurfsmustern

- überlegen, welches Entwurfsmuster **warum** Sinn macht

Aufgabe 4: Geometrify umstrukturieren

- Überlegungen aus Aufgabe 3 umsetzen

Tipps - 5. Übungsblatt

Aufgabe 5: GUI erweitern

- nochmal ServiceLoader $\Longrightarrow$ diesmal mit Primitiven

Denkt dran!

Abgabe

- Deadline am 5.7. um 12:00
- Aufgabe 1, 3 handschriftlich

Bis dann! (dann := 10.07.17)