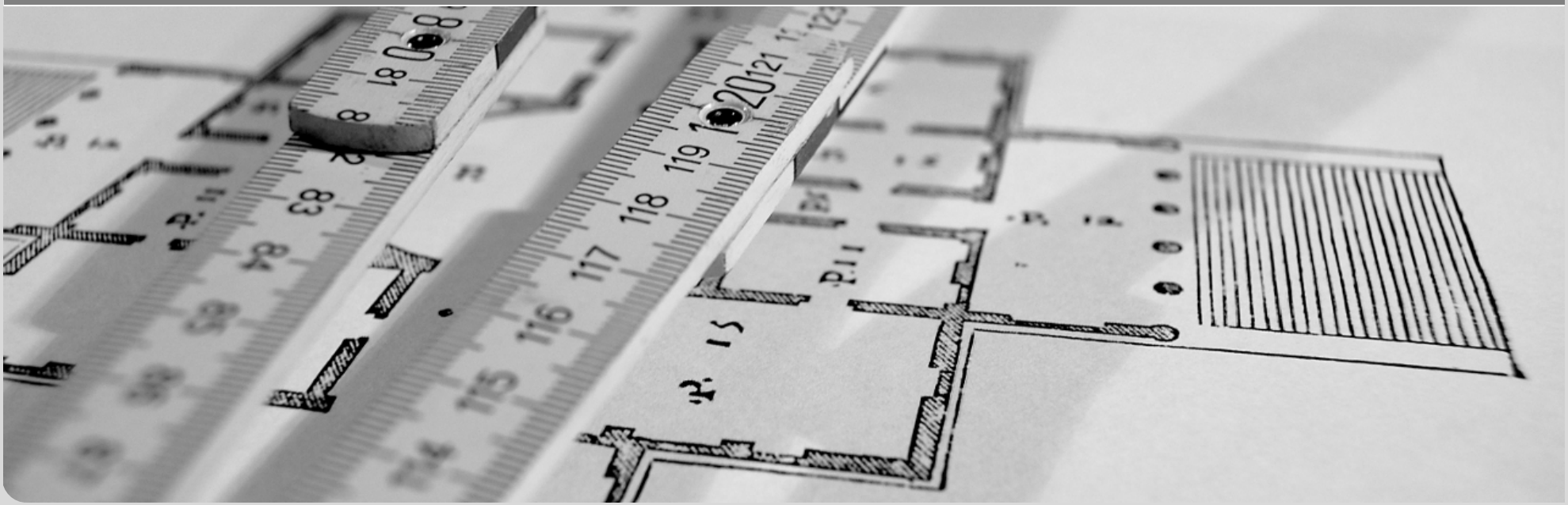
Karlsruher Institut für Technologie

Softwaretechnik 1 - 3. Tutorium

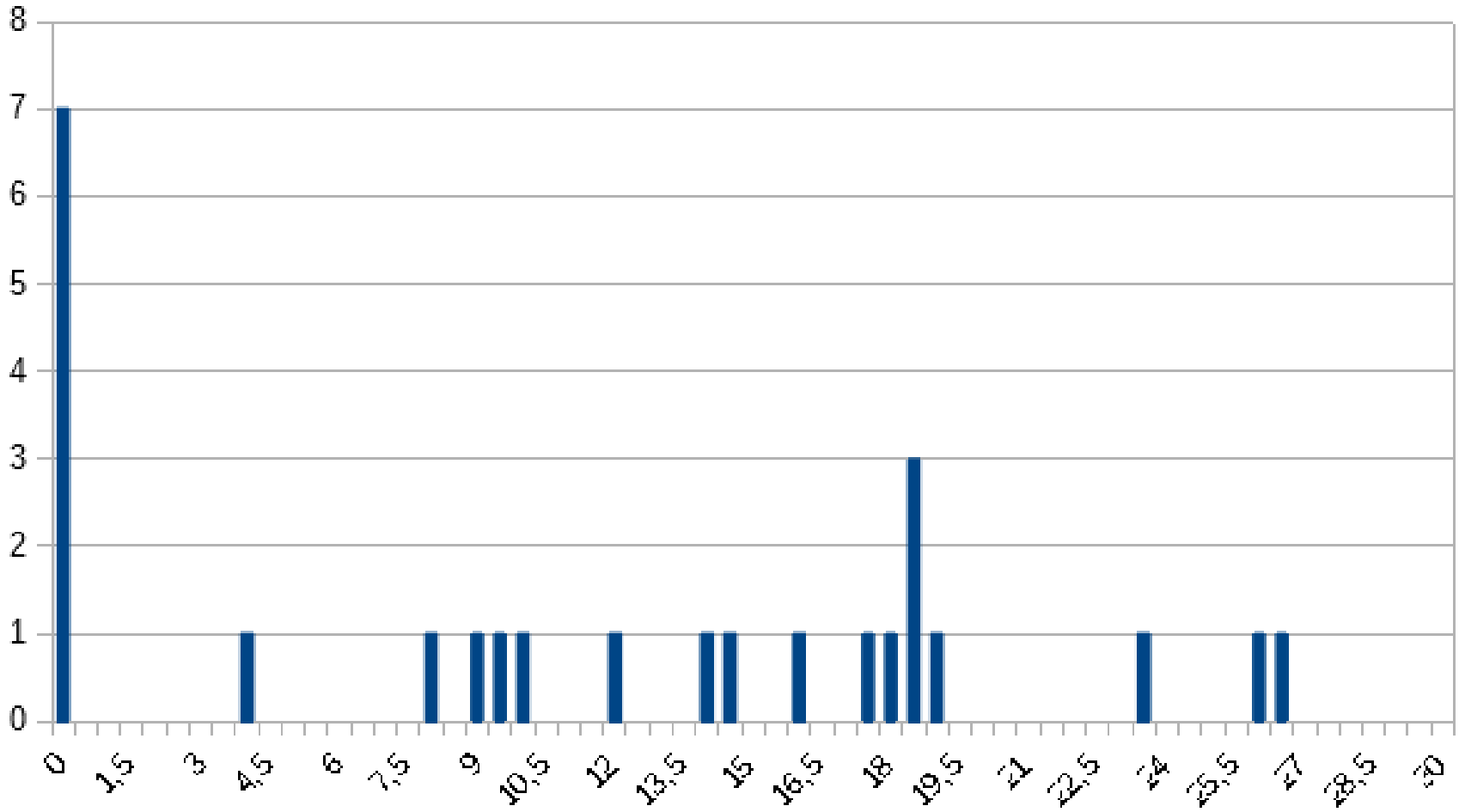
Tutorium 18

Felix Bachmann | 05.06.2018

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



3. Übungsblatt Statistik



∅ 11,3 bzw. 15,7 von 27+3

Häufige Fehler

Aufgabe 1 (Plug-In-Architektur: PluginManager + JmjrstPlugin)

Häufige Fehler

Aufgabe 1 (Plug-In-Architektur: PluginManager + JmjrstPlugin)

- Plugins anhand **Klassen**namen vergleichen, nicht getName()

Häufige Fehler

Aufgabe 1 (Plug-In-Architektur: PluginManager + JmjrstPlugin)

- Plugins anhand **Klassen**namen vergleichen, nicht getName()
- Strings auslagern (Konstanten oder Datei)

Häufige Fehler

Aufgabe 1 (Plug-In-Architektur: PluginManager + JmjrstPlugin)

- Plugins anhand **Klassen**namen vergleichen, nicht getName()
- Strings auslagern (Konstanten oder Datei)
- PluginManager gibt euch Iterable $\Longrightarrow$ nutzt Iterator
  - kein Casten, Kopieren in Liste

Häufige Fehler

Aufgabe 1 (Plug-In-Architektur: PluginManager + JmjrstPlugin)

- Plugins anhand **Klassen**namen vergleichen, nicht getName()
- Strings auslagern (Konstanten oder Datei)
- PluginManager gibt euch Iterable $\Longrightarrow$ nutzt Iterator
  - kein Casten, Kopieren in Liste
- orientiert euch nicht am JMJRST-Stil

Häufige Fehler

Aufgabe 2 (Plug-In)

Häufige Fehler

Aufgabe 2 (Plug-In)

- Prüfen auf *.png/*.jpg sollte case insensitive sein

Häufige Fehler

Aufgabe 2 (Plug-In)

- Prüfen auf *.png/*.jpg sollte case insensitive sein
- Anm.: MetainfServices tut manchmal nicht richtig (oft hilft `mvn clean package`)

Häufige Fehler

Aufgabe 2 (Plug-In)

- Prüfen auf *.png/*.jpg sollte case insensitive sein
- Anm.: MetainfServices tut manchmal nicht richtig (oft hilft `mvn clean package`)

Aufgabe 3 (iMage-Bundle)

- keine :D

Häufige Fehler

Aufgabe 4 (Aktivitätsdiagramm)

Häufige Fehler

Aufgabe 4 (Aktivitätsdiagramm)

- keine Partition verwendet

Häufige Fehler

Aufgabe 4 (Aktivitätsdiagramm)

- keine Partition verwendet
- Aktivitiäten = runde Ecken, Objekte = spitze Ecken

Häufige Fehler

Aufgabe 4 (Aktivitätsdiagramm)

- keine Partition verwendet
- Aktivitiäten = runde Ecken, Objekte = spitze Ecken
- denkt an die Rauten!

Häufige Fehler

Aufgabe 4 (Aktivitätsdiagramm)

- keine Partition verwendet
- Aktivitiäten = runde Ecken, Objekte = spitze Ecken
- denkt an die Rauten!
- [Bedingung]

Häufige Fehler

Aufgabe 4 (Aktivitätsdiagramm)

- keine Partition verwendet
- Aktivitiäten = runde Ecken, Objekte = spitze Ecken
- denkt an die Rauten!
- [Bedingung]
- verschachtelte Aktivitäten $\Longrightarrow$ irgendwo passender Kasten dazu

Häufige Fehler

Aufgabe 5 (Zustandsdiagramm)

Häufige Fehler

Aufgabe 5 (Zustandsdiagramm)

- Übergänge,Zustände vergessen

Häufige Fehler

Aufgabe 5 (Zustandsdiagramm)

- Übergänge,Zustände vergessen
- Notation parallel: DxG

Häufige Fehler

Aufgabe 5 (Zustandsdiagramm)

- Übergänge,Zustände vergessen
- Notation parallel: DxG
- komplettes Diagramm hinzeichnen für Äquivalenz

Häufige Fehler

Aufgabe 5 (Zustandsdiagramm)

- Übergänge,Zustände vergessen
- Notation parallel: DxG
- komplettes Diagramm hinzeichnen für Äquivalenz

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen
- Instanz-Kästen erst dann hinzeichnen, wenn Instanz auch existiert

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen
- Instanz-Kästen erst dann hinzeichnen, wenn Instanz auch existiert
- Selbstaufruf-Syntax

Aufgabe 7 (Testen mit Nachahmungen)

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen
- Instanz-Kästen erst dann hinzeichnen, wenn Instanz auch existiert
- Selbstaufruf-Syntax

Aufgabe 7 (Testen mit Nachahmungen)

- Substitutionsprinzip: fordert dass Objekte einer Unterklasse immer auch im Kontext der Oberklasse eingesetzt werden können

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen
- Instanz-Kästen erst dann hinzeichnen, wenn Instanz auch existiert
- Selbstaufruf-Syntax

Aufgabe 7 (Testen mit Nachahmungen)

- Substitutionsprinzip: fordert dass Objekte einer Unterklasse immer auch im Kontext der Oberklasse eingesetzt werden können
- Varianz war kein Problem, da Signatur gleich

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen
- Instanz-Kästen erst dann hinzeichnen, wenn Instanz auch existiert
- Selbstaufruf-Syntax

Aufgabe 7 (Testen mit Nachahmungen)

- Substitutionsprinzip: fordert dass Objekte einer Unterklasse immer auch im Kontext der Oberklasse eingesetzt werden können
- Varianz war kein Problem, da Signatur gleich
- Problem war Verhalten, schwächere Nachbedingung

Häufige Fehler

Aufgabe 6 (Sequenzdiagramm)

- bzgl. Konstruktor sind VL-Folien etwas blöd
- asynchron vs. synchron (Pfeilspitzen sind wichtig!)
- nicht statische Instanzen unterstreichen
- Instanz-Kästen erst dann hinzeichnen, wenn Instanz auch existiert
- Selbstaufruf-Syntax

Aufgabe 7 (Testen mit Nachahmungen)

- Substitutionsprinzip: fordert dass Objekte einer Unterklasse immer auch im Kontext der Oberklasse eingesetzt werden können
- Varianz war kein Problem, da Signatur gleich
- Problem war Verhalten, schwächere Nachbedingung

Wo sind wir?

- die ersten 2 Phasen des Wasserfallmodells sind geschafft

Wo sind wir?

- die ersten 2 Phasen des Wasserfallmodells sind geschafft
  $\Longrightarrow$ Welche waren das nochmal?

Wo sind wir?

- die ersten 2 Phasen des Wasserfallmodells sind geschafft
  $\Longrightarrow$ Welche waren das nochmal? Planung, Definition!

Wo sind wir?

- die ersten 2 Phasen des Wasserfallmodells sind geschafft
  $\Longrightarrow$ Welche waren das nochmal? Planung, Definition!
  $\Longrightarrow$ Dokumente?

Wo sind wir?

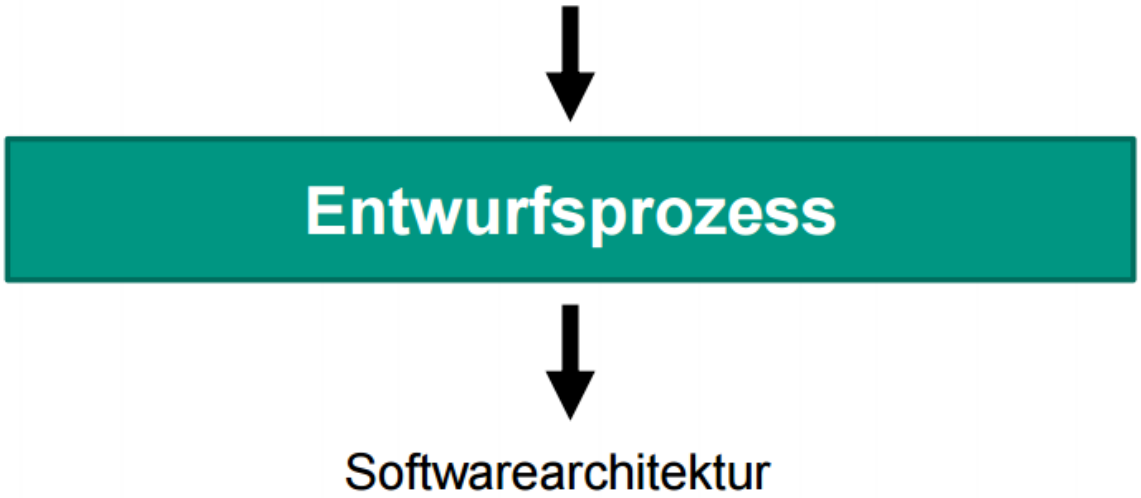
- die ersten 2 Phasen des Wasserfallmodells sind geschafft
  $\Longrightarrow$ Welche waren das nochmal? Planung, Definition!
  $\Longrightarrow$ Dokumente? Lastenheft, Pflichtenheft (+ andere...)

Wo sind wir?

- die ersten 2 Phasen des Wasserfallmodells sind geschafft
  $\Longrightarrow$ Welche waren das nochmal? Planung, Definition!
  $\Longrightarrow$ Dokumente? Lastenheft, Pflichtenheft (+ andere...)
- jetzt: Entwurf!

Wozu Entwurf?

- Pflichtenheft (einschl. Modelle)
- Konzept Benutzungsoberfläche
- Benutzerhandbuch + Hilfekonzept



Softwarearchitektur ist Grundlage für Implementierung!

Abgrenzung Definition vs. Entwurf

- Definition: **Was** ist zu implementieren?

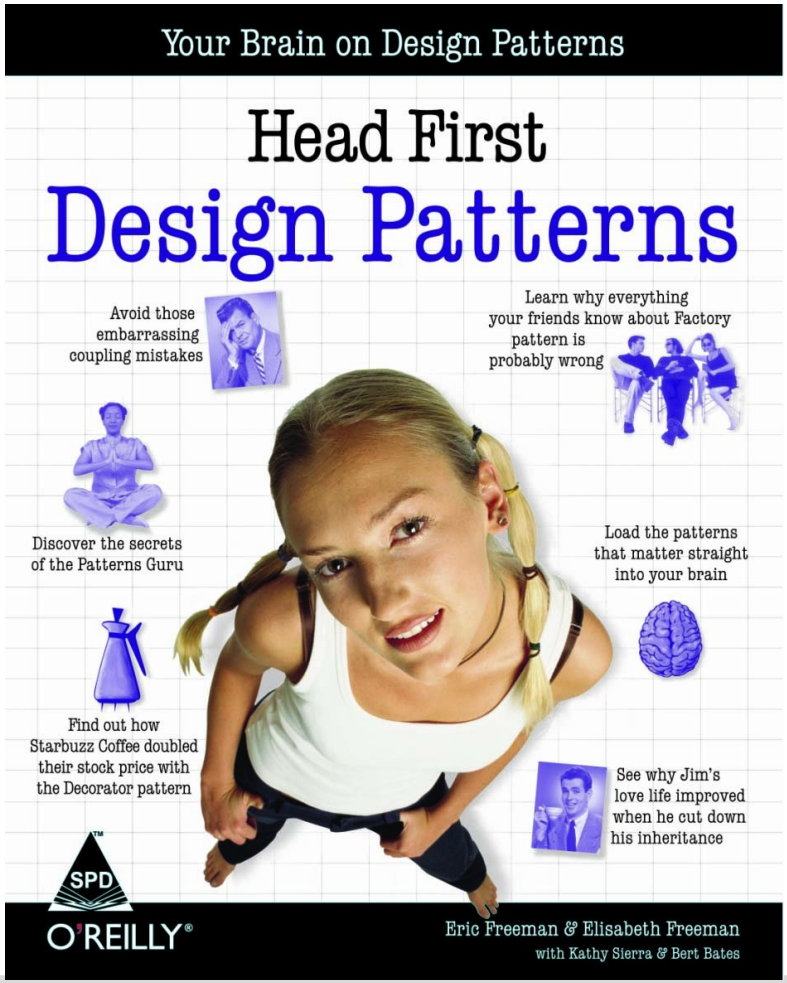
Abgrenzung Definition vs. Entwurf

- Definition: **Was** ist zu implementieren?
- Entwurf: **Wie** ist das System zu implementieren?

Empfehlenswerte Literatur (wirklich!)

knapp 700 Seiten

$\Longrightarrow$ als interaktives Nachschlagewerk, falls man bestimmte Muster nicht versteht



Was sind Entwurfsmuster?

Entwurfsmuster

Ein Software-Entwurfsmuster beschreibt eine Familie von Lösungen für ein Software-Entwurfsproblem.

Was sind Entwurfsmuster?

Entwurfsmuster

Ein Software-Entwurfsmuster beschreibt eine Familie von Lösungen für ein Software-Entwurfsproblem.

- schematische Klassendiagramme zur Lösung von häufig auftretenden Problemen

Was sind Entwurfsmuster?

Entwurfsmuster

Ein Software-Entwurfsmuster beschreibt eine Familie von Lösungen für ein Software-Entwurfsproblem.

- schematische Klassendiagramme zur Lösung von häufig auftretenden Problemen
- Wiederverwendung von Entwurfswissen $\Longrightarrow$ Rad nicht neu erfinden!

Was sind Entwurfsmuster?

Entwurfsmuster

Ein Software-Entwurfsmuster beschreibt eine Familie von Lösungen für ein Software-Entwurfsproblem.

- schematische Klassendiagramme zur Lösung von häufig auftretenden Problemen
- Wiederverwendung von Entwurfswissen $\Longrightarrow$ Rad nicht neu erfinden!

Wozu Entwurfsmuster?

- erleichtern Kommunikation

Wozu Entwurfsmuster?

- erleichtern Kommunikation
- erleichtern “gute“ Entwürfe und das Schreiben von wartbarem/erweiterbarem Code

Geheimnisprinzip

Geheimnis- / Kapselungsprinzip

Jedes Modul verbirgt eine wichtige Entwurfsentscheidung hinter einer wohldefinierten Schnittstelle, die sich bei einer Änderung der Entscheidung nicht mit ändert.

Geheimnisprinzip

Geheimnis- / Kapselungsprinzip

Jedes Modul verbirgt eine wichtige Entwurfsentscheidung hinter einer wohldefinierten Schnittstelle, die sich bei einer Änderung der Entscheidung nicht mit ändert.

Warum eigentlich?

lokale Änderungen sollen sich nicht auf andere Teile auswirken

$\Longrightarrow$ weniger Fehler und Arbeit

Beispiel?

Geheimnisprinzip

Geheimnis- / Kapselungsprinzip

Jedes Modul verbirgt eine wichtige Entwurfsentscheidung hinter einer wohldefinierten Schnittstelle, die sich bei einer Änderung der Entscheidung nicht mit ändert.

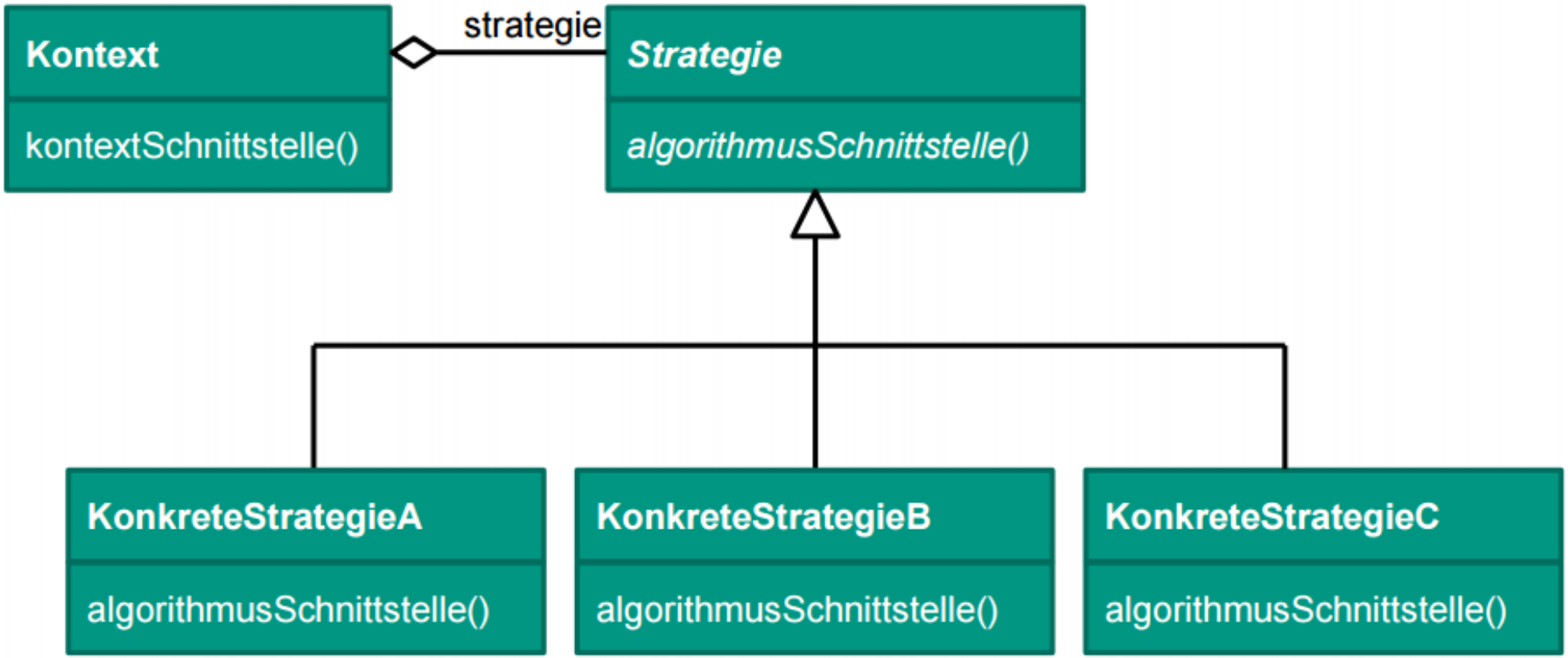
Warum eigentlich?

lokale Änderungen sollen sich nicht auf andere Teile auswirken
$\Longrightarrow$ weniger Fehler und Arbeit

Beispiel? $\Longrightarrow$ private Attribute mit get()- und set()-Methoden

Vorgriff: Entwurfsmuster Strategie

- Ziel: Algorithmen kapseln, austauschbar machen
- wird in vielen Entwurfsmustern verwendet



Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren.

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren.
  wahr

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren.
  wahr
- Das Strategiemuster erfüllt das Geheimnisprinzip, indem es Datenstrukturen, die in einer konkreten Strategie enthalten sind, vor dem Klienten verbirgt.

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren. wahr
- Das Strategiemuster erfüllt das Geheimnisprinzip, indem es Datenstrukturen, die in einer konkreten Strategie enthalten sind, vor dem Klienten verbirgt. wahr

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren. wahr
- Das Strategiemuster erfüllt das Geheimnisprinzip, indem es Datenstrukturen, die in einer konkreten Strategie enthalten sind, vor dem Klienten verbirgt. wahr
- Das Muster Strategie kapselt austauschbares Verhalten und verwendet Delegierung, um zu entscheiden, welches Verhalten verwendet wird.

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren. wahr
- Das Strategiemuster erfüllt das Geheimnisprinzip, indem es Datenstrukturen, die in einer konkreten Strategie enthalten sind, vor dem Klienten verbirgt. wahr
- Das Muster Strategie kapselt austauschbares Verhalten und verwendet Delegierung, um zu entscheiden, welches Verhalten verwendet wird. wahr

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren. wahr
- Das Strategiemuster erfüllt das Geheimnisprinzip, indem es Datenstrukturen, die in einer konkreten Strategie enthalten sind, vor dem Klienten verbirgt. wahr
- Das Muster Strategie kapselt austauschbares Verhalten und verwendet Delegierung, um zu entscheiden, welches Verhalten verwendet wird. wahr
- Das Hinzufügen einer neuen konkreten Strategie erfordert keine Änderung existierender konkreter Strategien.

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Das Entwurfsmuster Strategie bietet die Möglichkeit, eine Klasse mit einer von mehreren möglichen Verhaltensweisen zu konfigurieren. wahr
- Das Strategiemuster erfüllt das Geheimnisprinzip, indem es Datenstrukturen, die in einer konkreten Strategie enthalten sind, vor dem Klienten verbirgt. wahr
- Das Muster Strategie kapselt austauschbares Verhalten und verwendet Delegierung, um zu entscheiden, welches Verhalten verwendet wird. wahr
- Das Hinzufügen einer neuen konkreten Strategie erfordert keine Änderung existierender konkreter Strategien. wahr

Kategorien der Entwurfsmuster

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

Entkopplungs-Muster

- übergeordnetes Ziel: System in Teile aufspalten, die unabhängig voneinander sind
  $\Longrightarrow$ Teile austauschbar bzw. veränderbar

Adapter

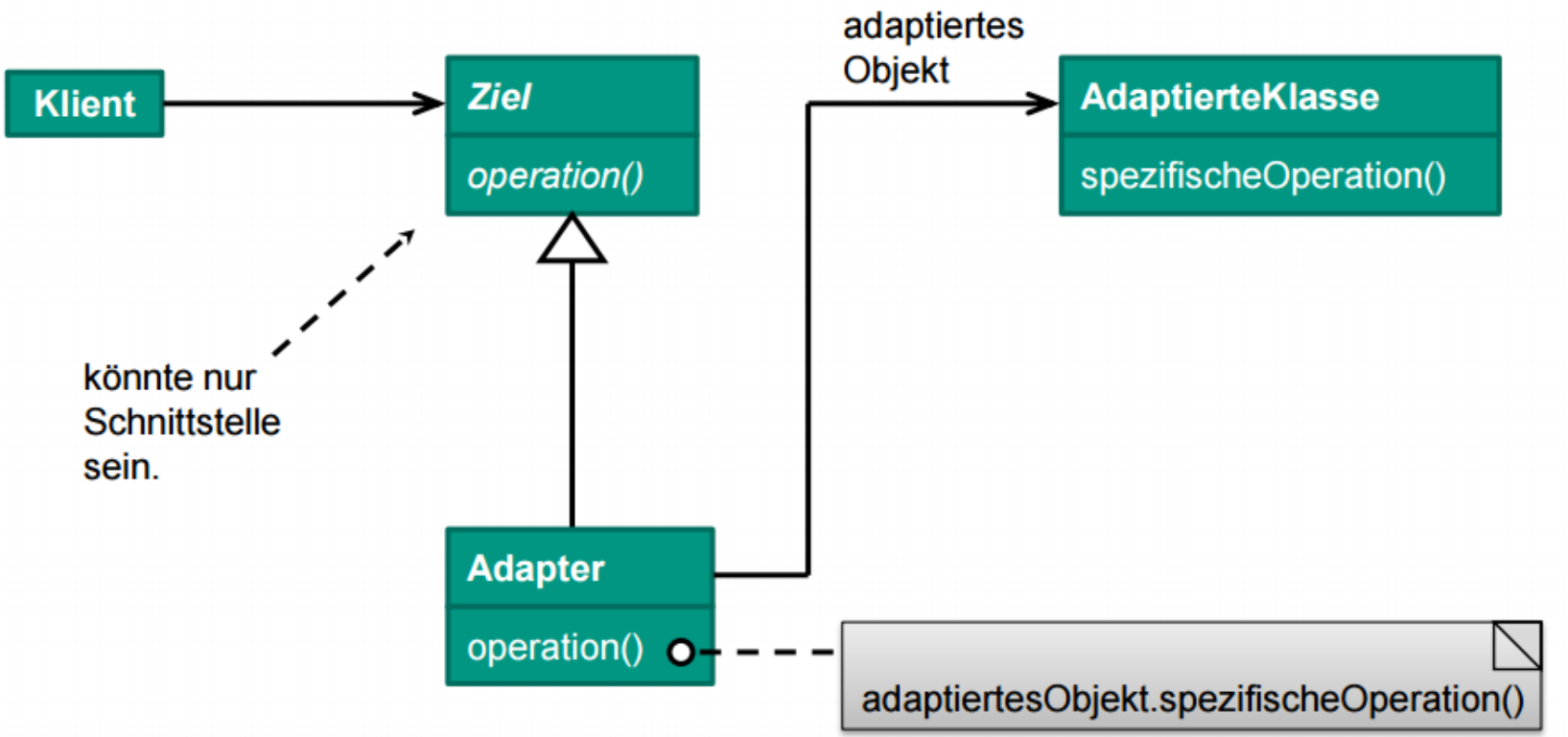
Problem

- Klassen mit inkompatiblen Schnittstellen, die wir aber zusammen benutzen wollen
- Schnittstellen nicht änderbar (z.B. externe Bibliotheken)

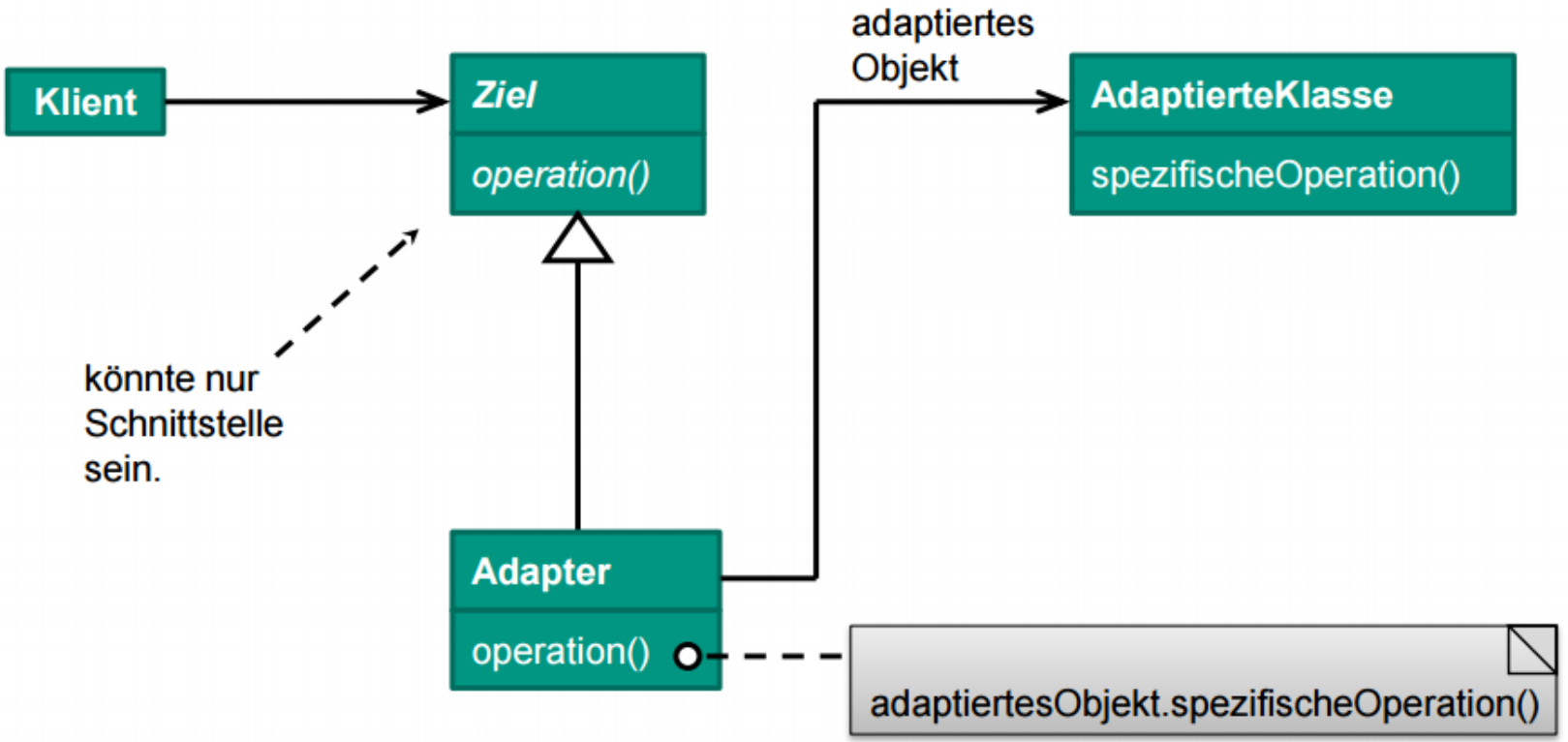
Adapter

Problem

- Klassen mit inkompatiblen Schnittstellen, die wir aber zusammen benutzen wollen
- Schnittstellen nicht änderbar (z.B. externe Bibliotheken)



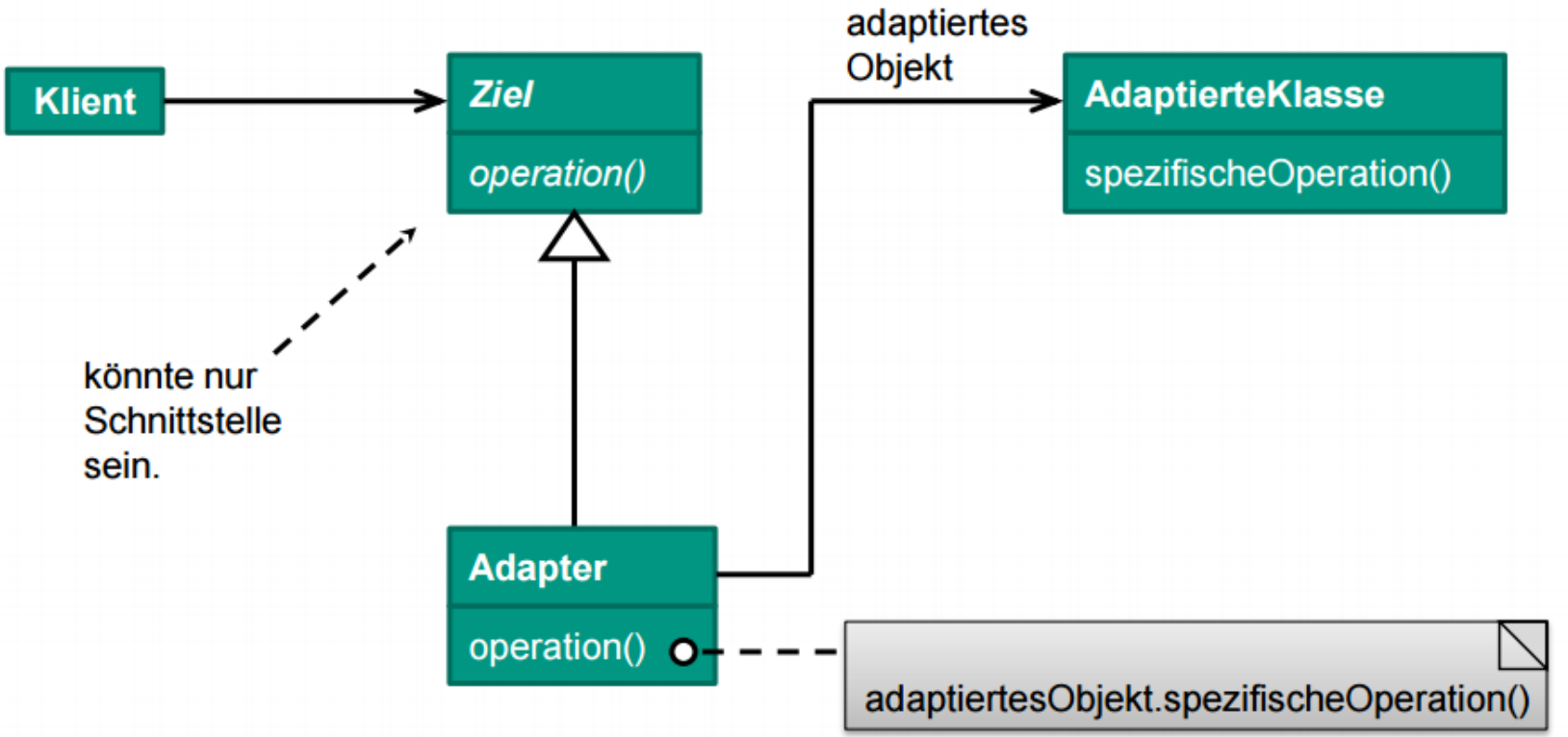
Adapter (Objektadapter)



Wir sind bei Entkopplung-Mustern, Preisfrage:

Wo ist hier die Entkopplung?

Adapter (Objektadapter)

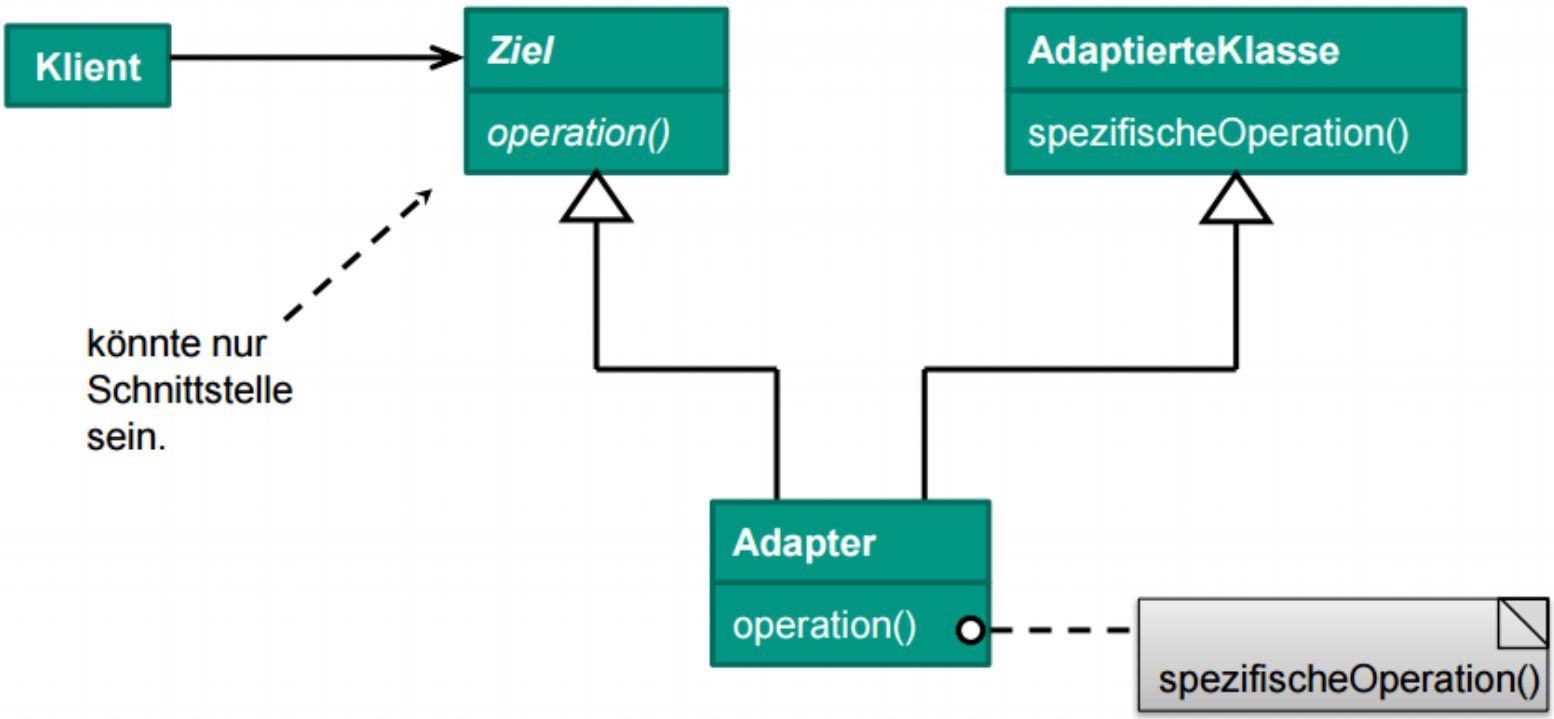


Wir sind bei Entkopplung-Mustern, Preisfrage:

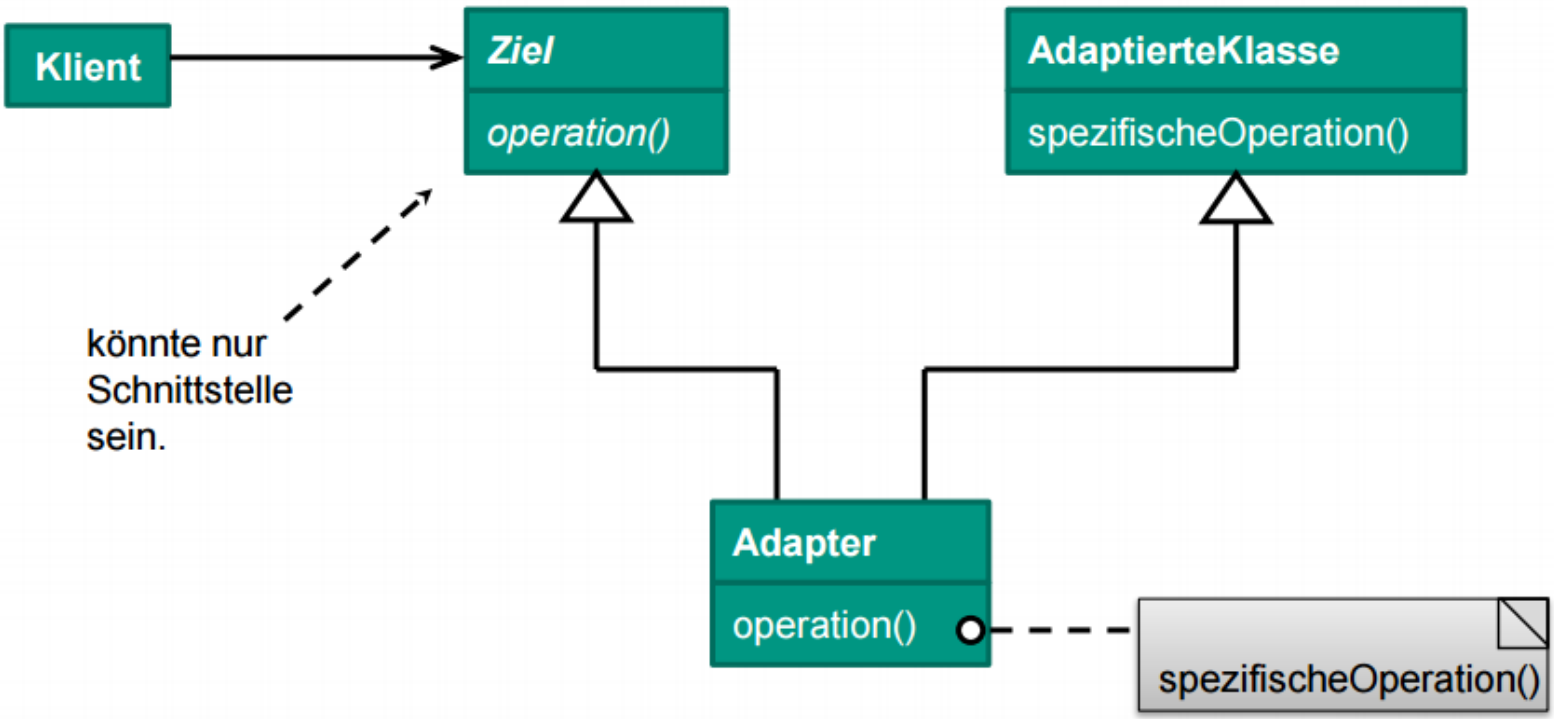
Wo ist hier die Entkopplung?

der Klient ist von der adaptierten Klasse entkoppelt $\Longrightarrow$ austauschbar

Adapter - Alternative (Klassenadapter)

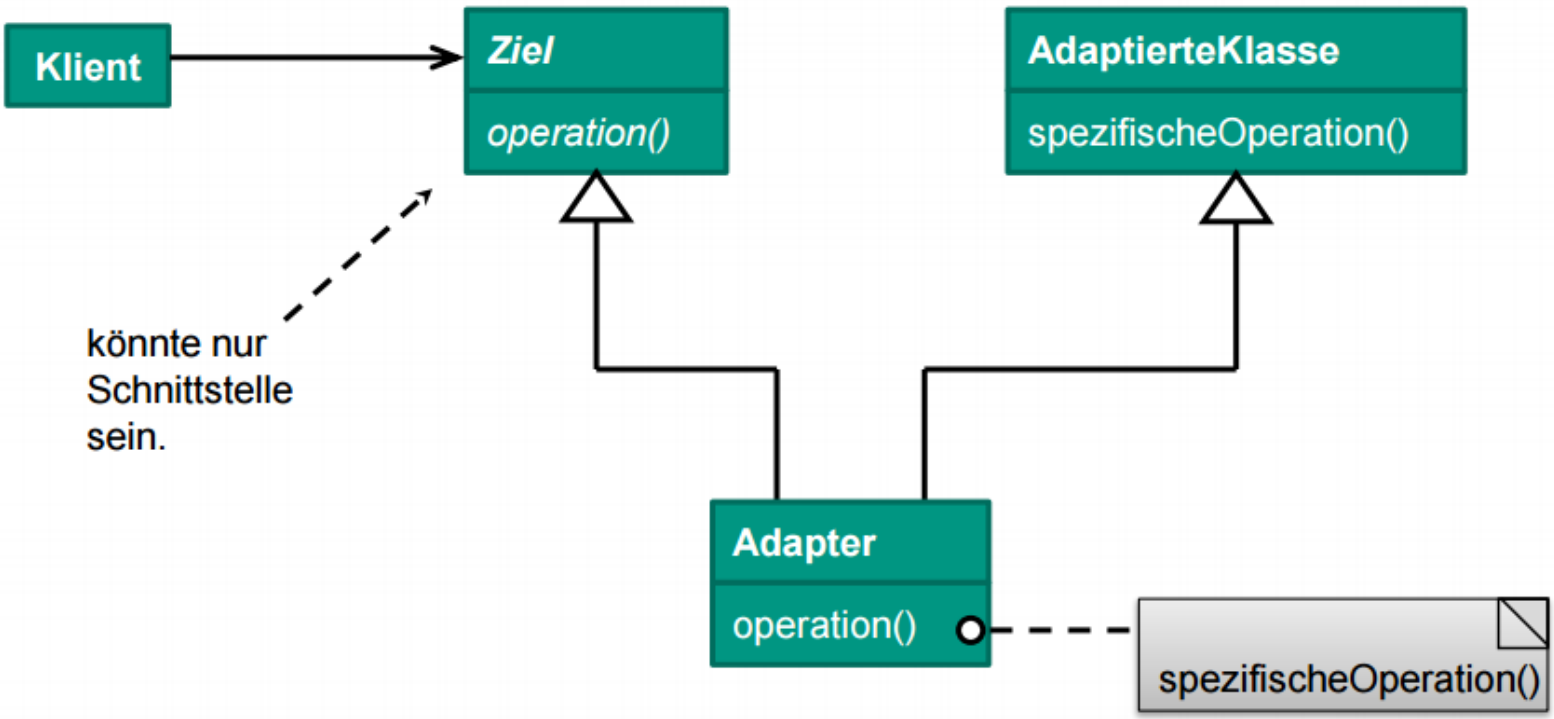


Adapter - Alternative (Klassenadapter)



Was für ein Problem bekommt ihr, wenn ihr das auf einem ÜB implementieren müsst?

Adapter - Alternative (Klassenadapter)



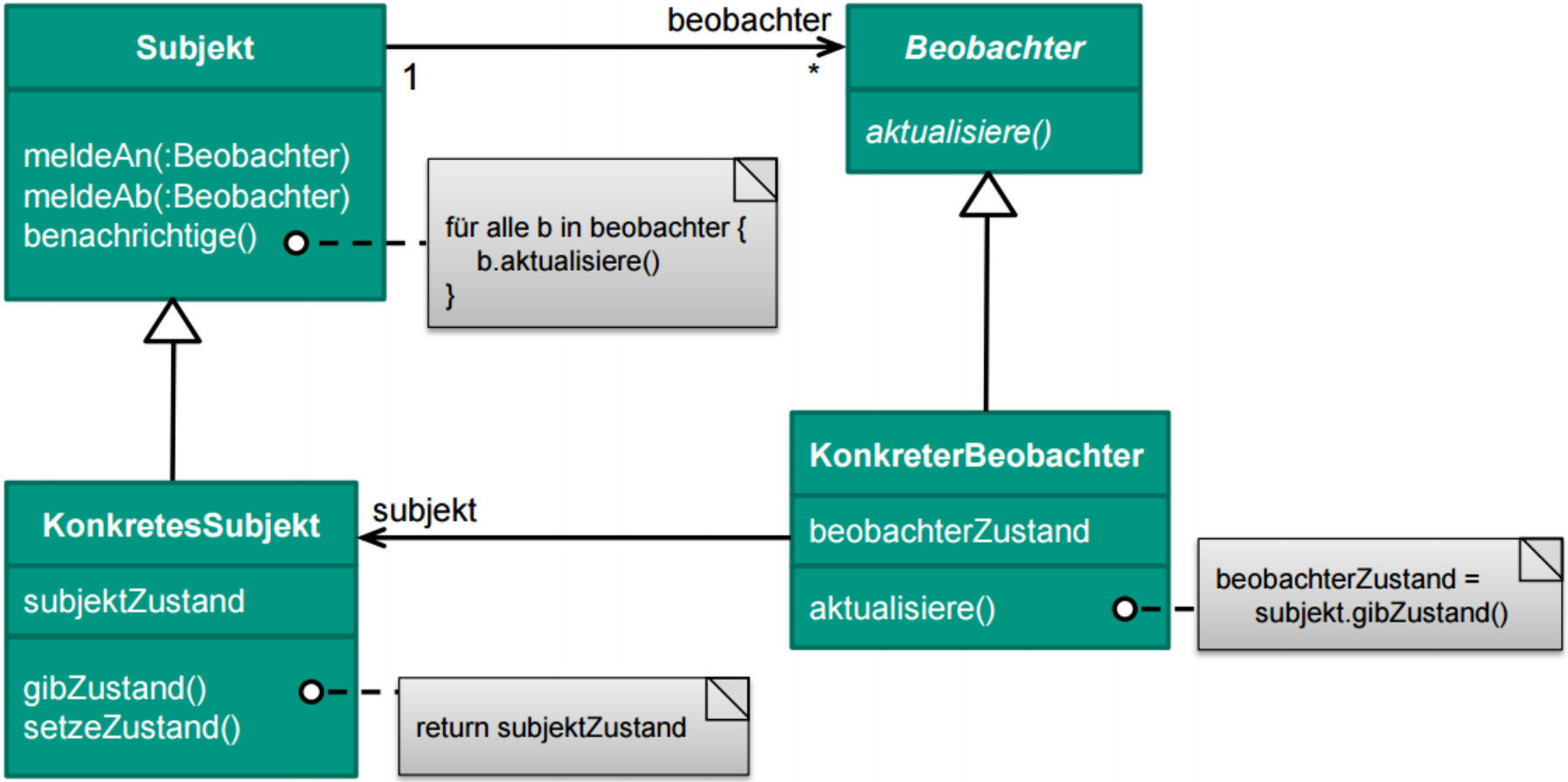
Was für ein Problem bekommt ihr, wenn ihr das auf einem ÜB implementieren müsst?

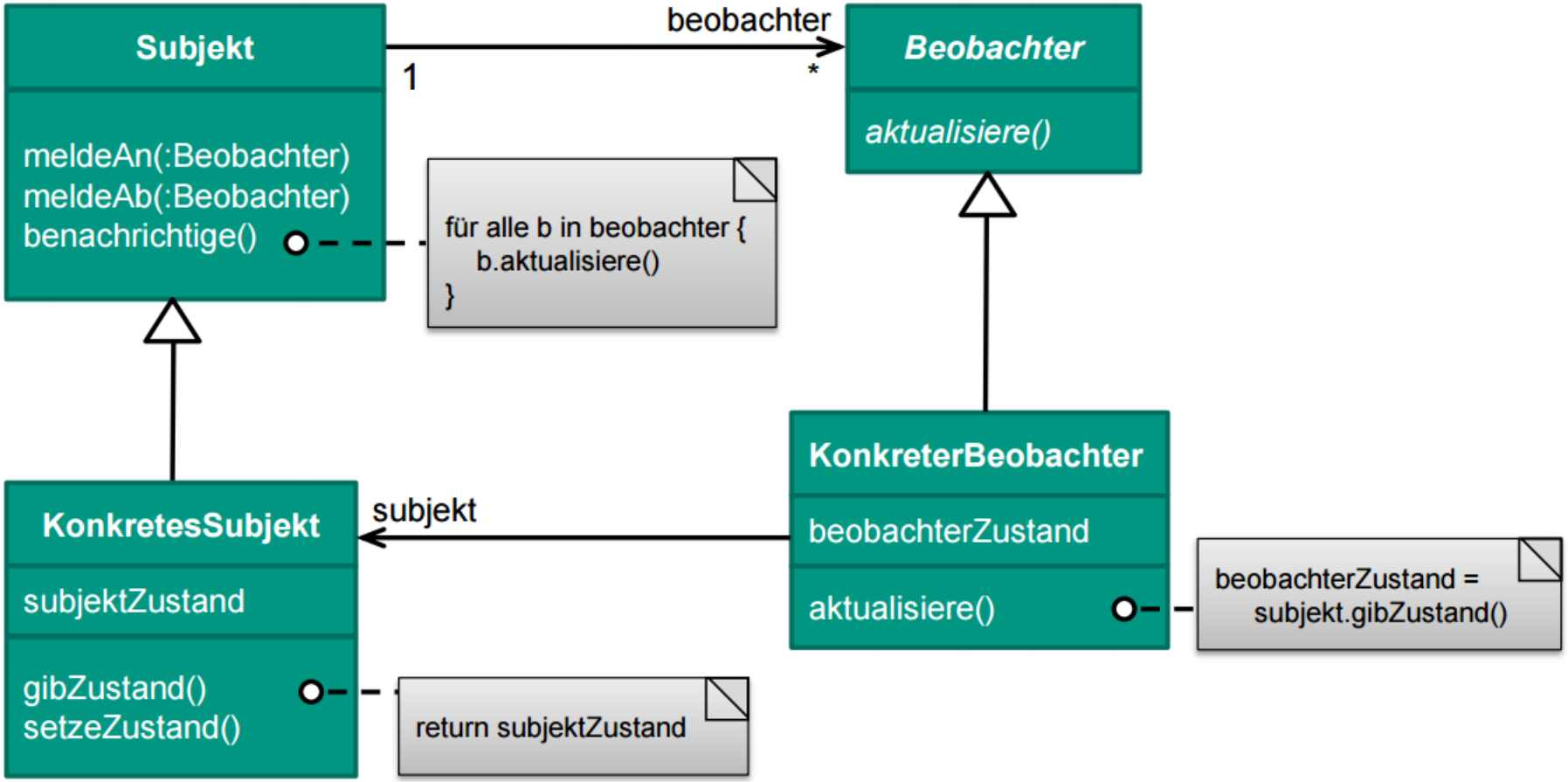
$\Longrightarrow$ keine Mehrfachvererbung in Java!

Beobachter/Observer: abstrakt

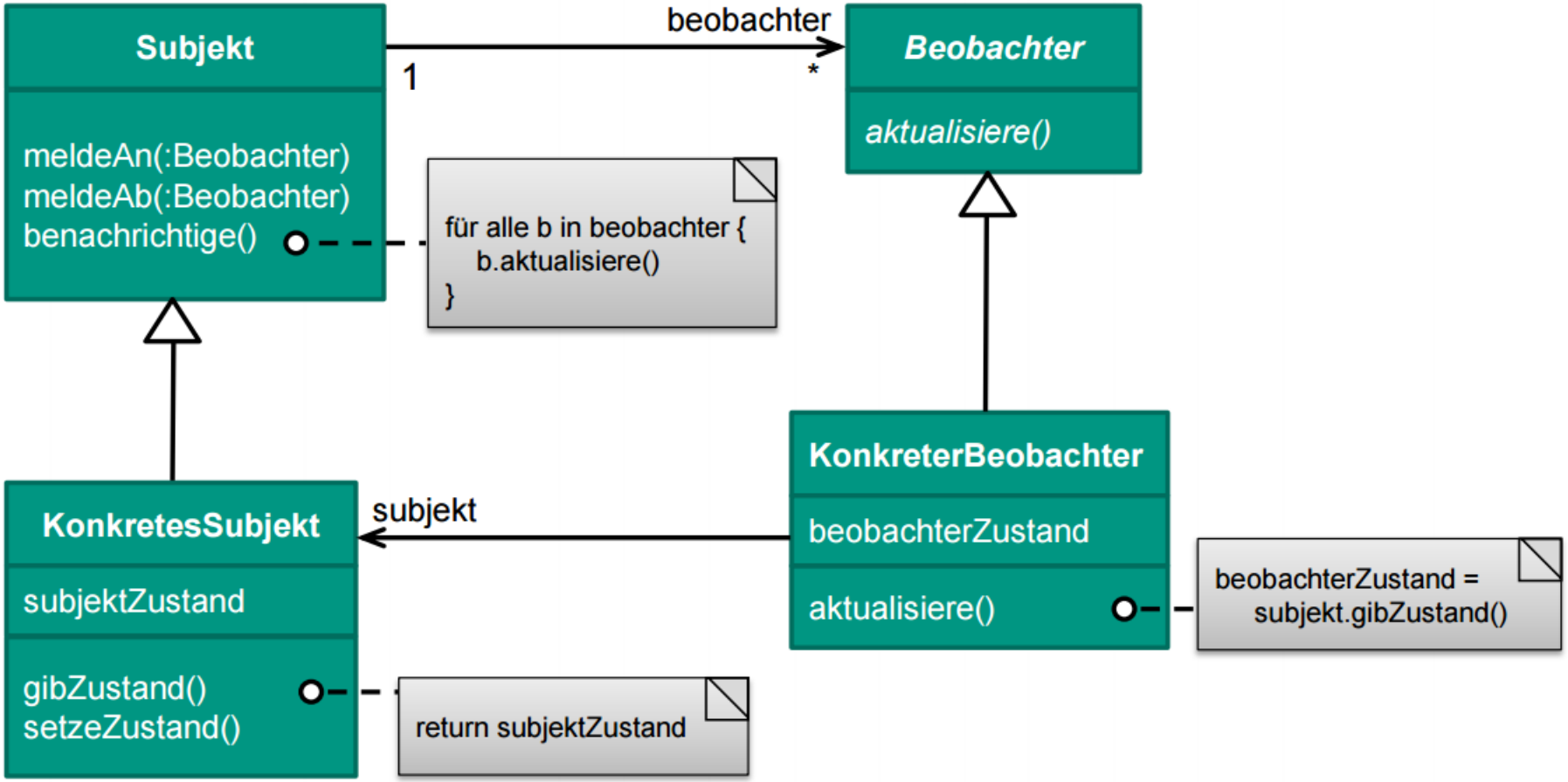
Problem

- ein Subjekt, viele Beobachter
- Subjekt ändert Zustand $\Longrightarrow$ Beobachter machen “irgendwas”


Subjekt
beobachter
1
*
Beobachter
aktualisiere()
meldeAn(:Beobachter)
meldeAb(:Beobachter)
benachrichtige()
für alle b in beobachter {
b.aktualisiere()
}
KonkreterBeobachter
beobachterZustand
aktualisiere()
beobachterZustand =
subjekt.gibZustand()
KonkretesSubjekt
subjekt
subjektZustand
gibZustand()
setzeZustand()
return subjektZustand

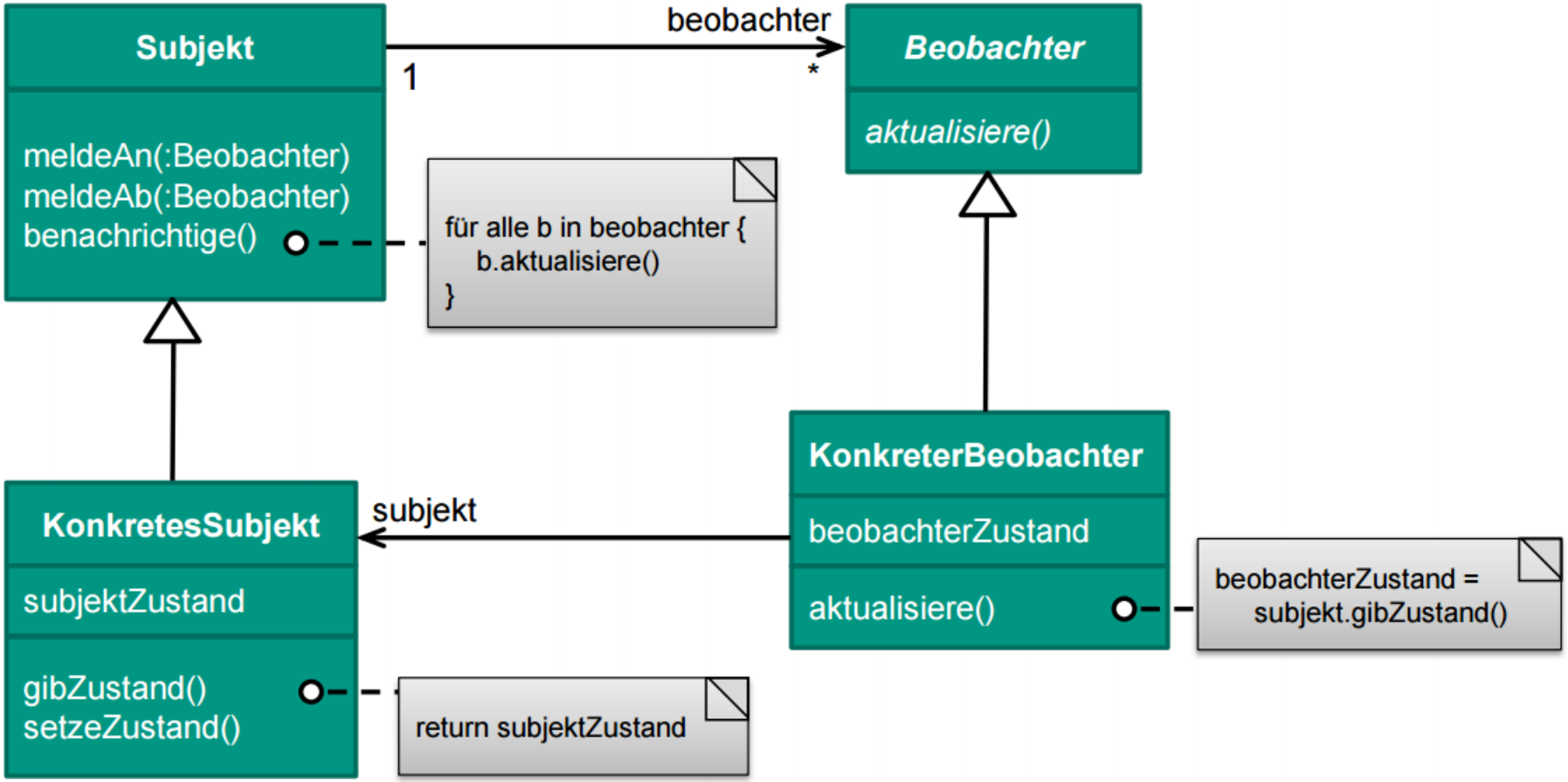


Entkopplung?



Entkopplung?

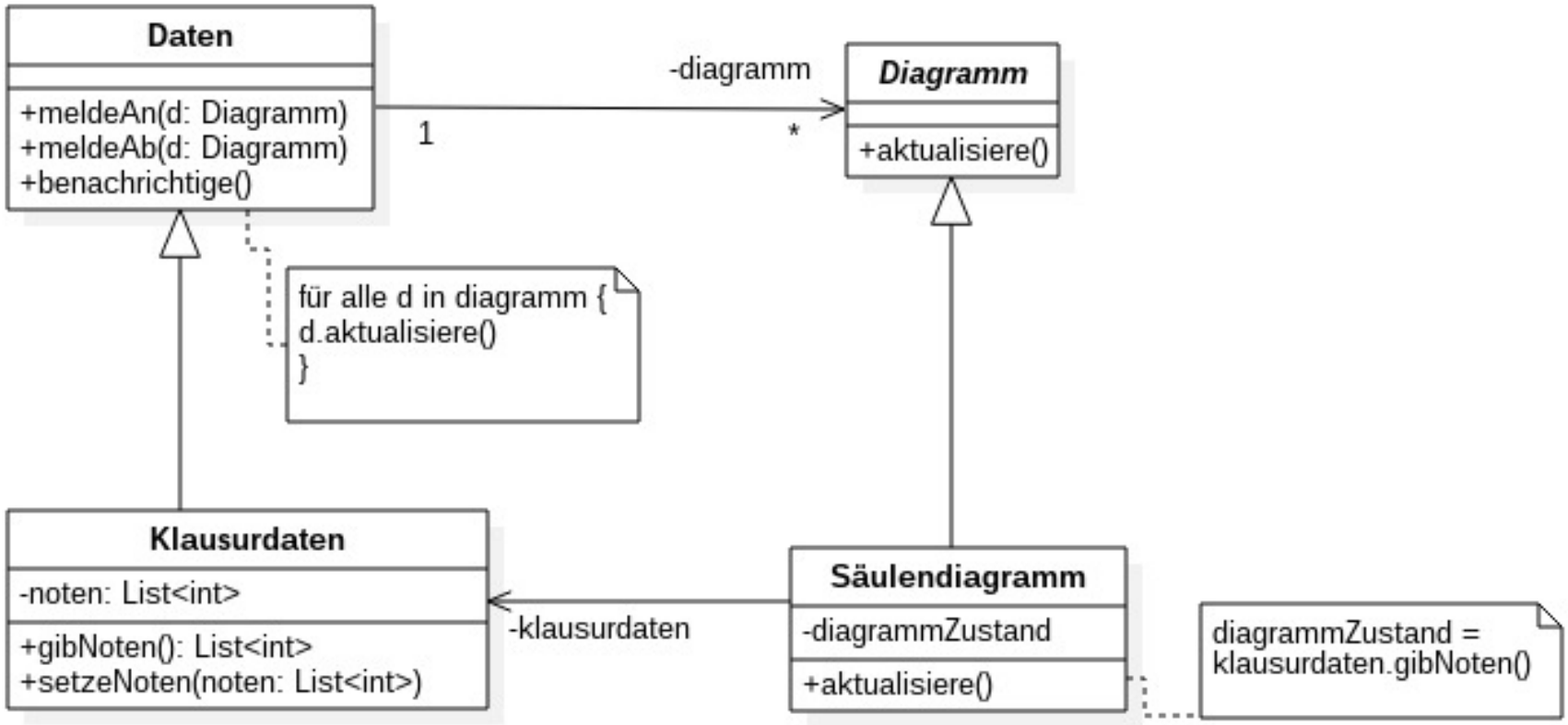
- jeder Beobachter definiert, was bei Benachrichtigung passiert, Subjekt kriegt davon nichts mit



Entkopplung?

- jeder Beobachter definiert, was bei Benachrichtigung passiert, Subjekt kriegt davon nichts mit
- zur Laufzeit änderbar: Anzahl der Beobachter

Beobachter/Observer: am Beispiel



Iterator

Problem

- wollen über Datenstruktur iterieren + Operationen ausführen

Iterator

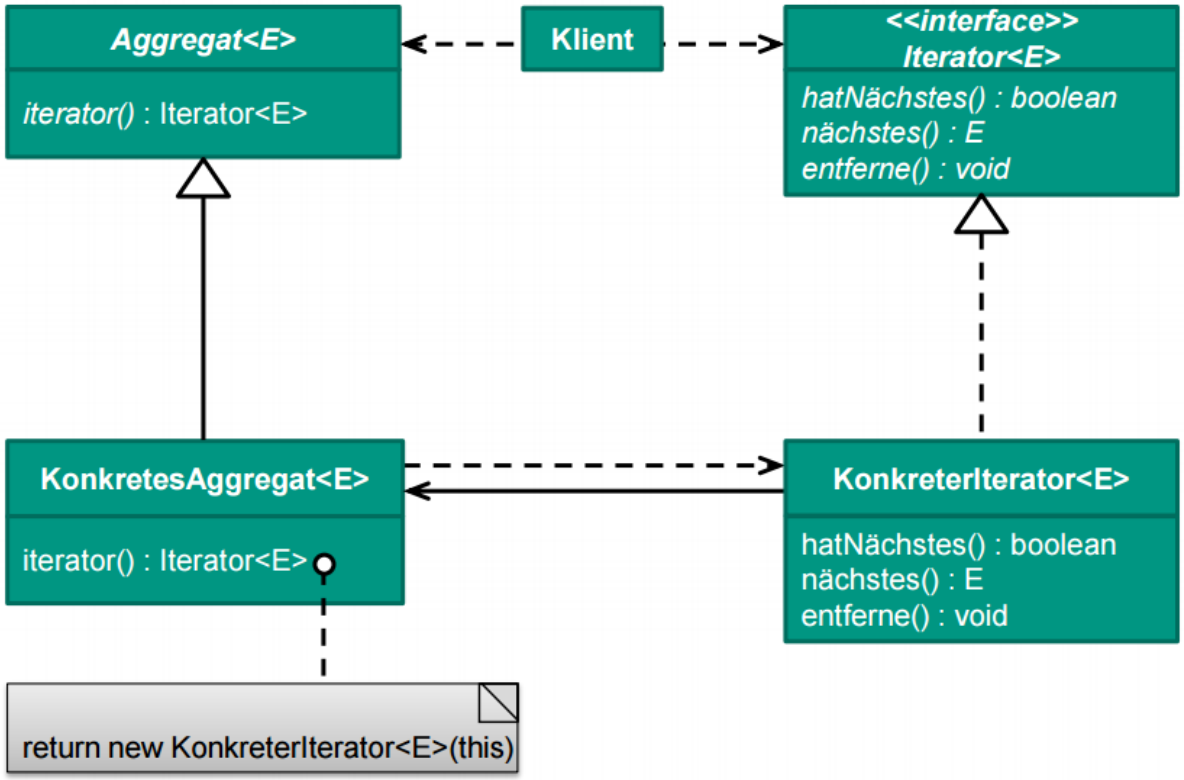
Problem

- wollen über Datenstruktur iterieren + Operationen ausführen
- das Ganze ohne Kentniss des internen Aufbaus der Datenstruktur

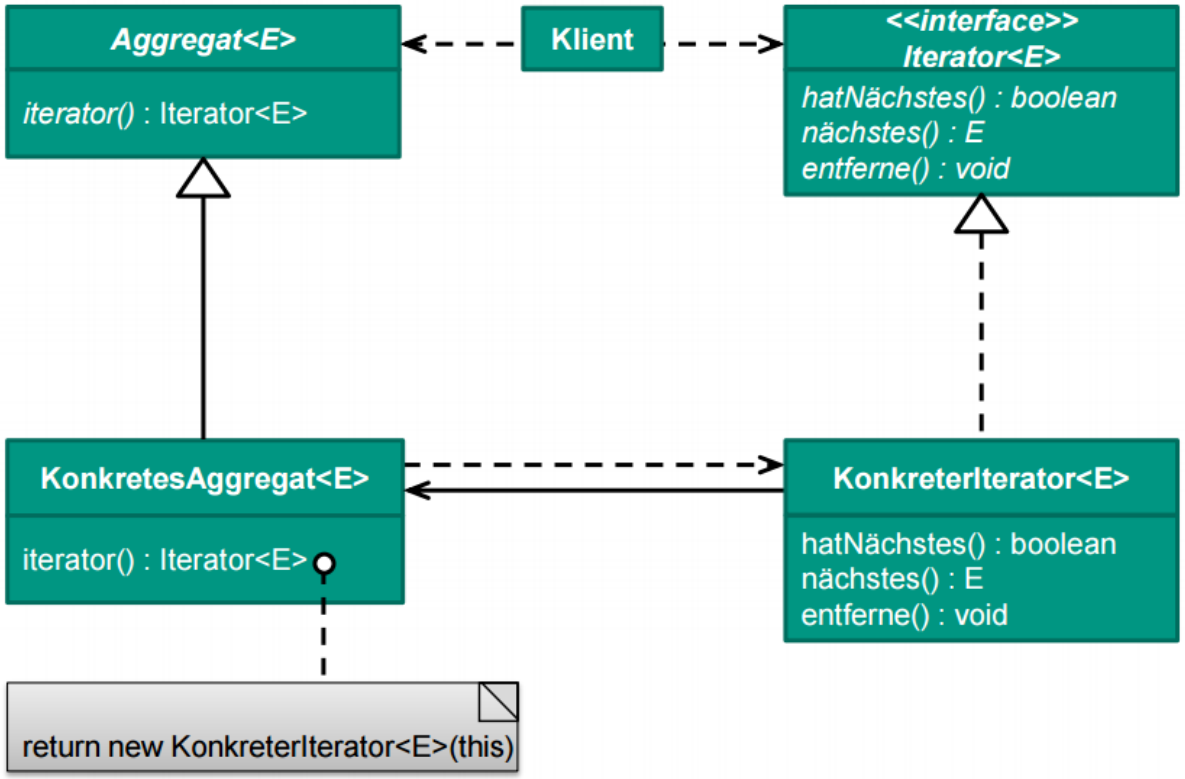
Iterator

Problem

- wollen über Datenstruktur iterieren + Operationen ausführen
- das Ganze ohne Kentniss des internen Aufbaus der Datenstruktur

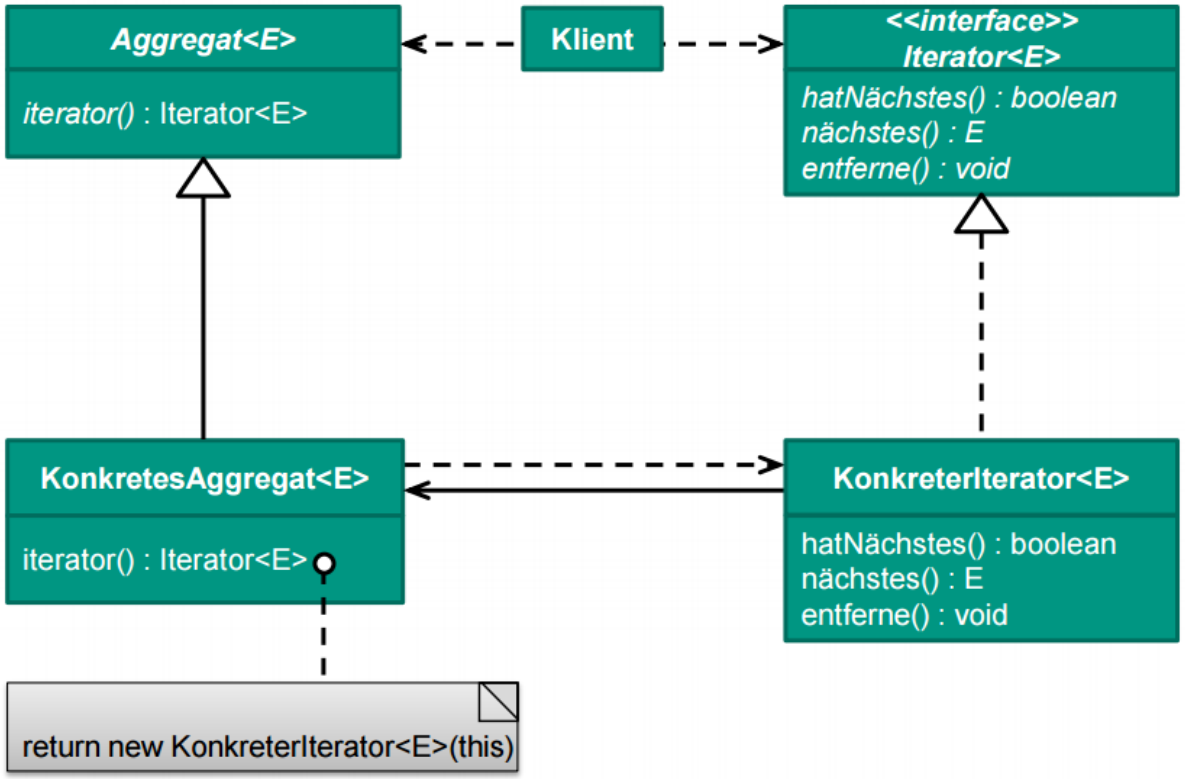


Iterator



Entkopplung?

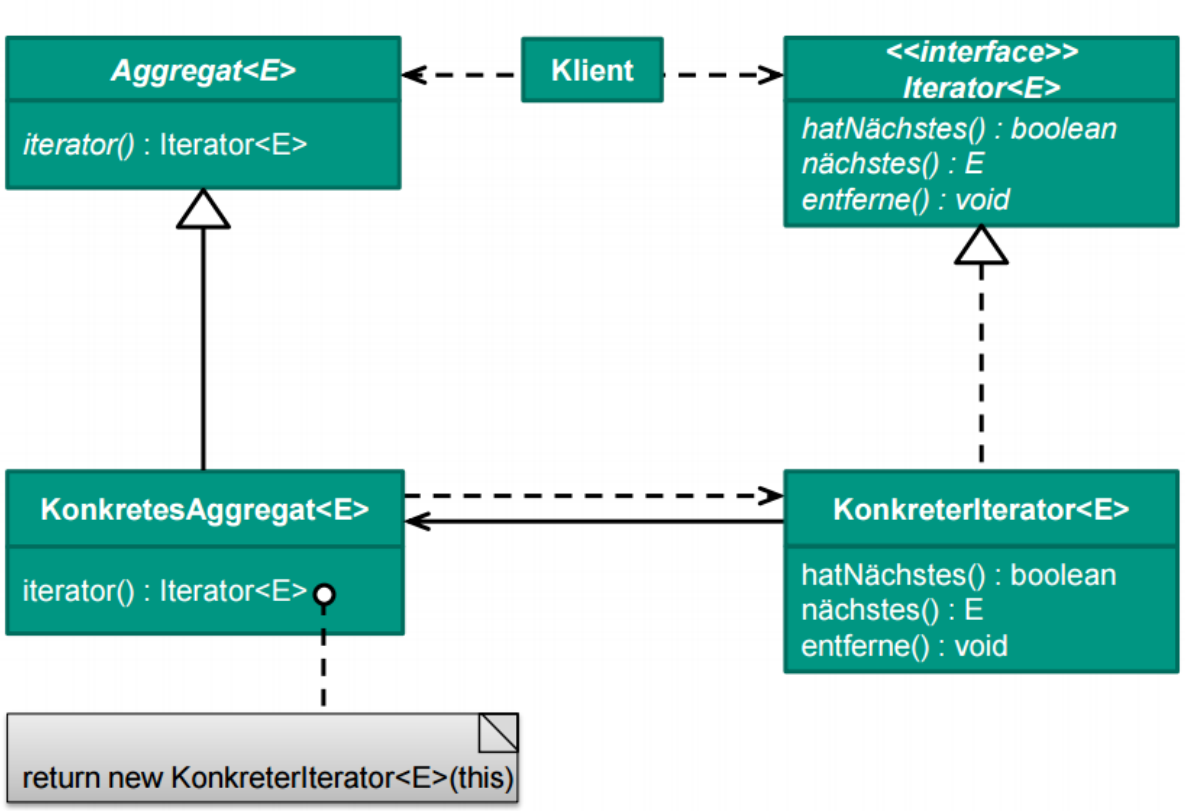
Iterator



Entkopplung?

- Klient benutzt nur Methoden der Schnittstelle auf dem konkreten Iterator
  $\Longrightarrow$ Implementierung austauschbar

Iterator



Beispiel in Java: list.iterator();

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Klienten können mithilfe des Iterator-Musters Sammlungen von Objekten und einzelne Objekte auf die gleiche Weise behandeln.

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Klienten können mithilfe des Iterator-Musters Sammlungen von Objekten und einzelne Objekte auf die gleiche Weise behandeln.
  falsch

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Klienten können mithilfe des Iterator-Musters Sammlungen von Objekten und einzelne Objekte auf die gleiche Weise behandeln.
  falsch
- Das Entwurfsmuster Iterator ist den Variantenmustern zuzuordnen.

Quiz (Ankreuzaufgaben aus Klausuren)

Wahr oder falsch?

- Klienten können mithilfe des Iterator-Musters Sammlungen von Objekten und einzelne Objekte auf die gleiche Weise behandeln.
  falsch
- Das Entwurfsmuster Iterator ist den Variantenmustern zuzuordnen.
  falsch

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

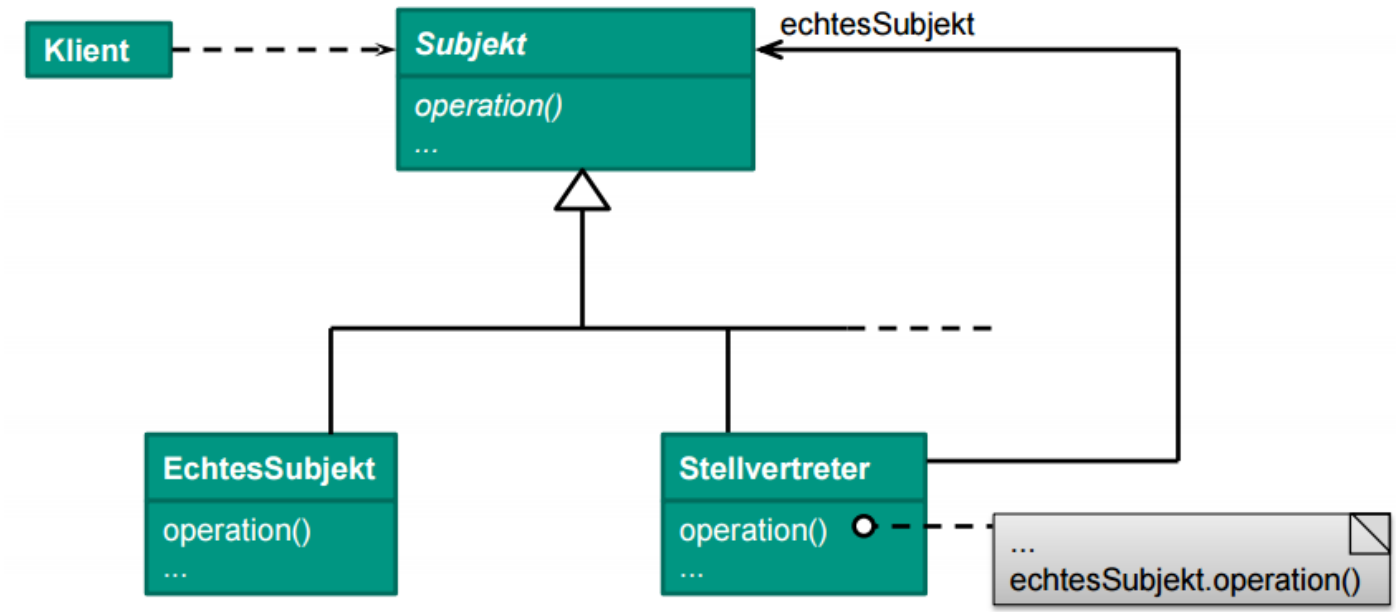
  $\Longrightarrow$ Stellvertreter macht Zugriffskontrolle

Stellvertreter

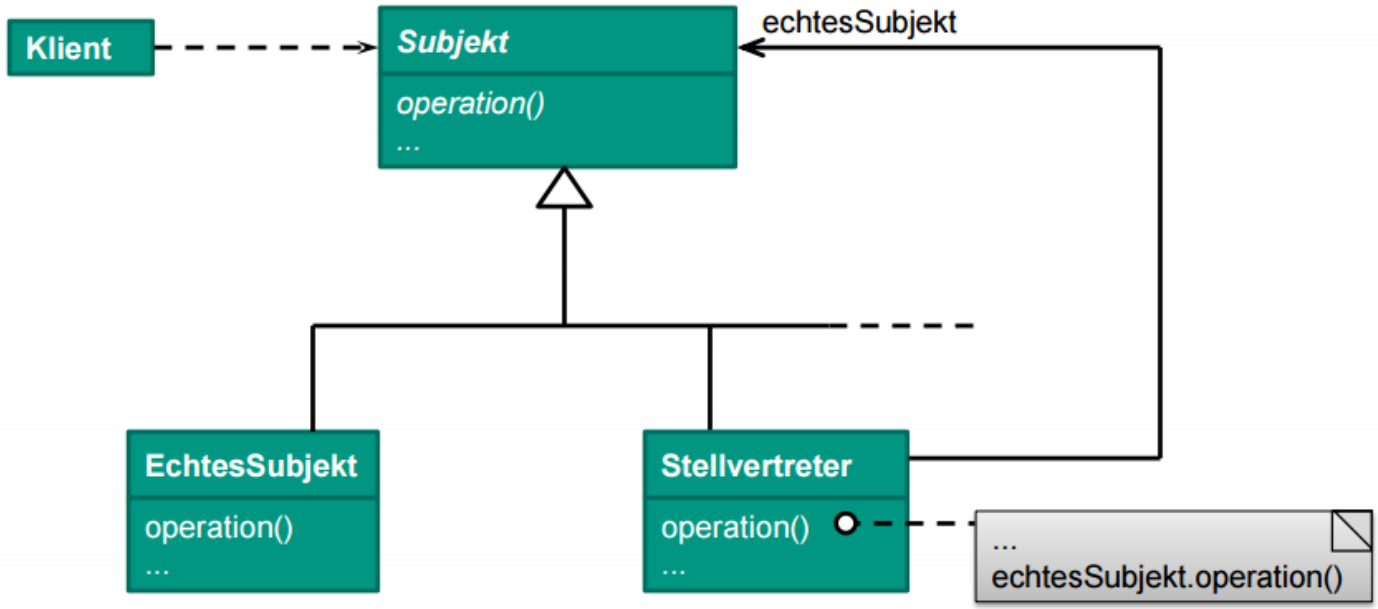
Problem

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

  $\Longrightarrow$ Stellvertreter macht Zugriffskontrolle

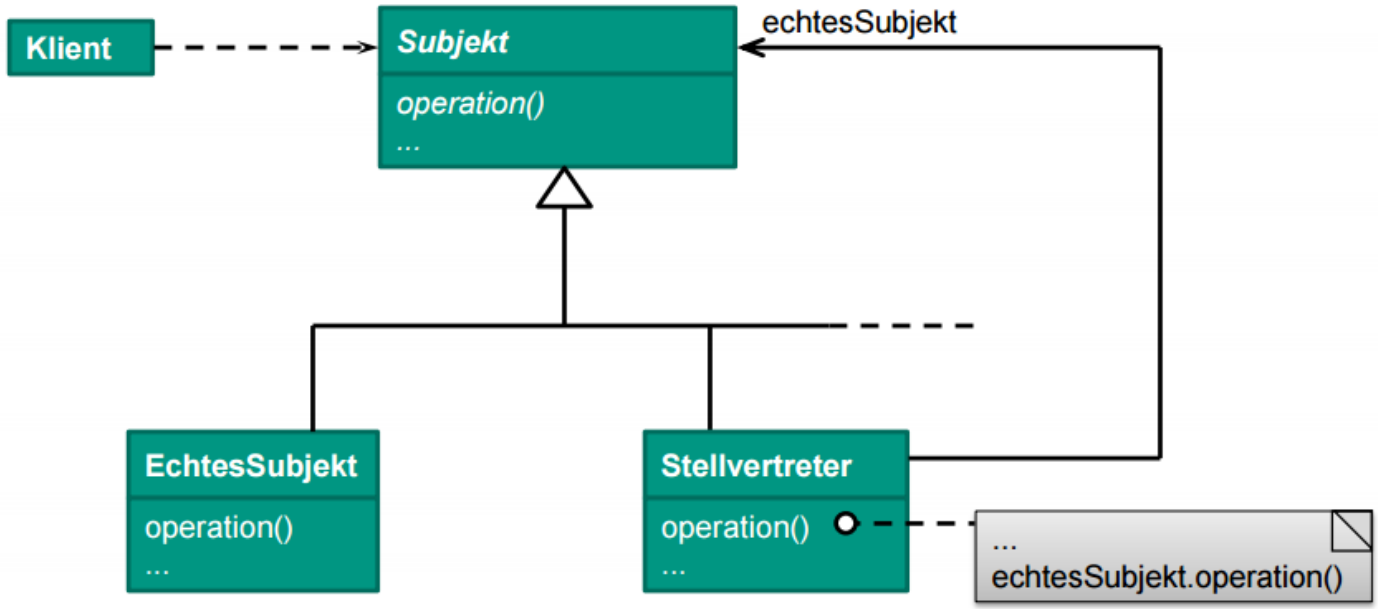


Stellvertreter



Entkopplung?

Stellvertreter



Entkopplung?

- Klient hat keinen direkten Zugriff auf das echte Subjekt

Vermittler

Problem

- mehrere voneinander abhängige Objekte

Vermittler

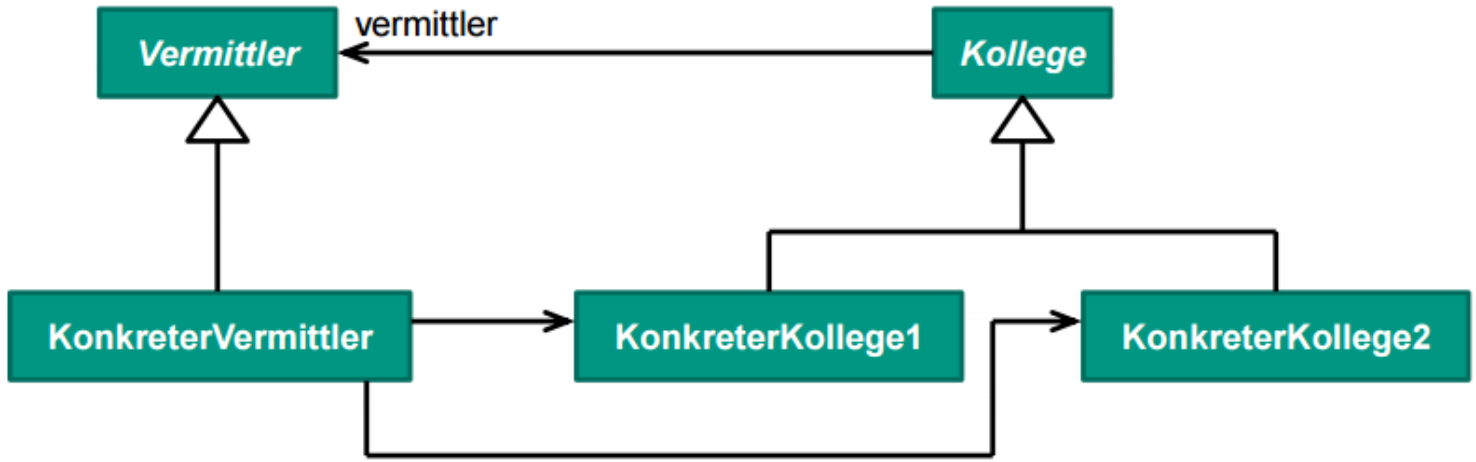
Problem

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

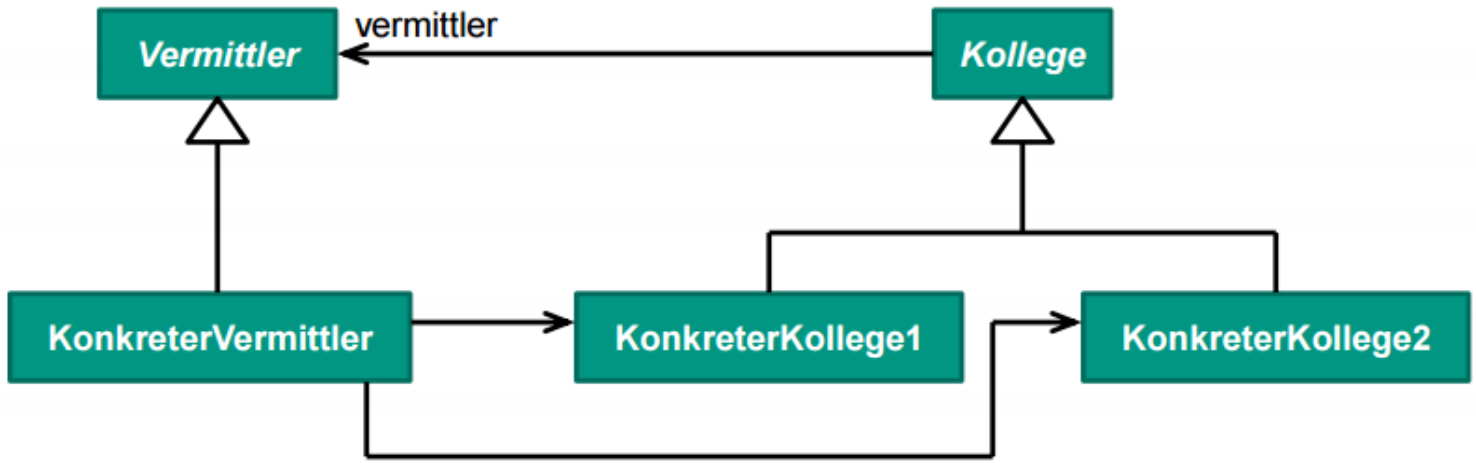
Vermittler

Problem

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

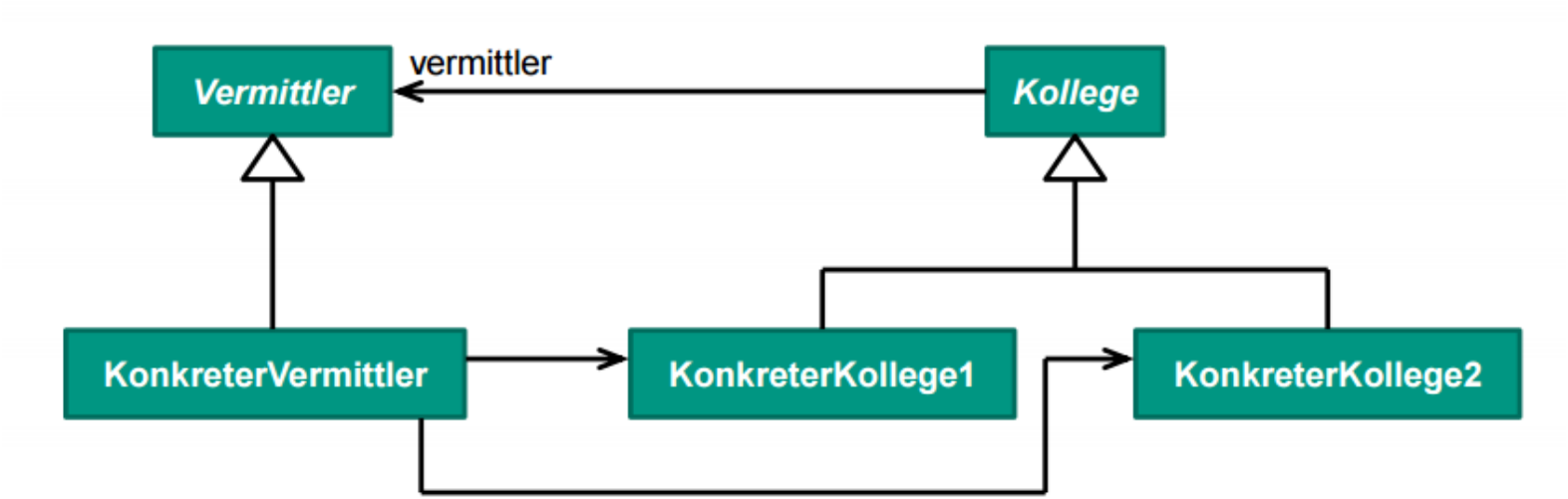


Vermittler



Entkopplung?

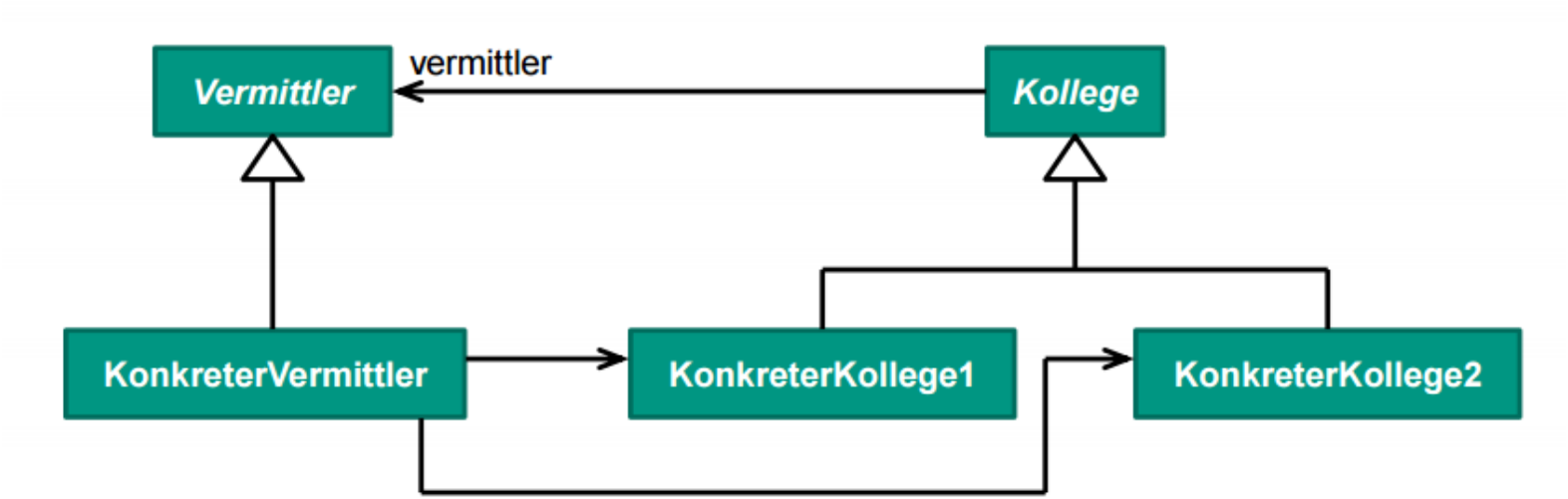
Vermittler



Entkopplung?

- Kollegen kennen sich nicht direkt

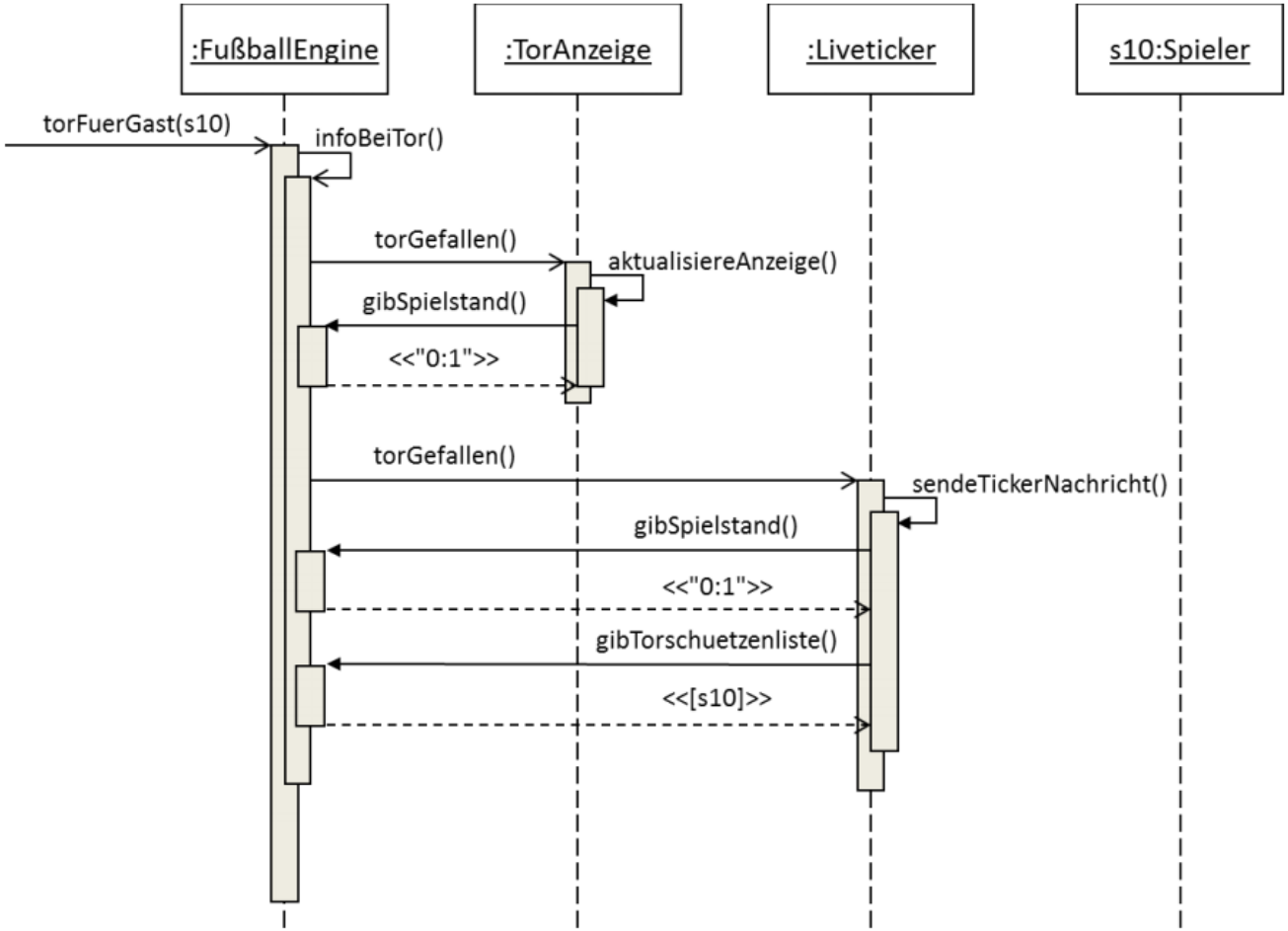
Vermittler



Entkopplung?

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

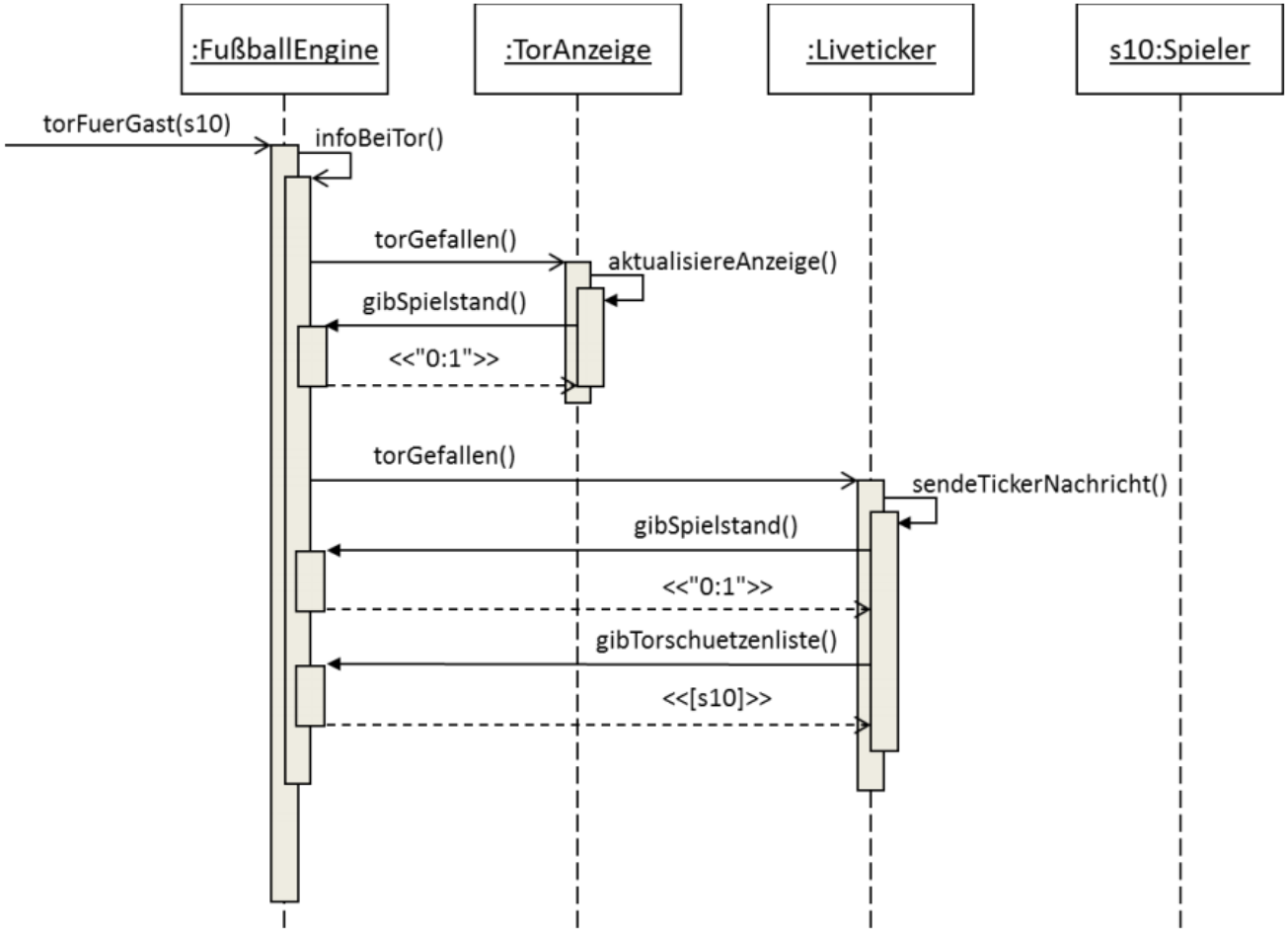
Klausuraufgabe (Hauptklausur SS 2012)



Aufgabe 1

Welches Entwurfsmuster erkennen Sie in diesem Diagramm?

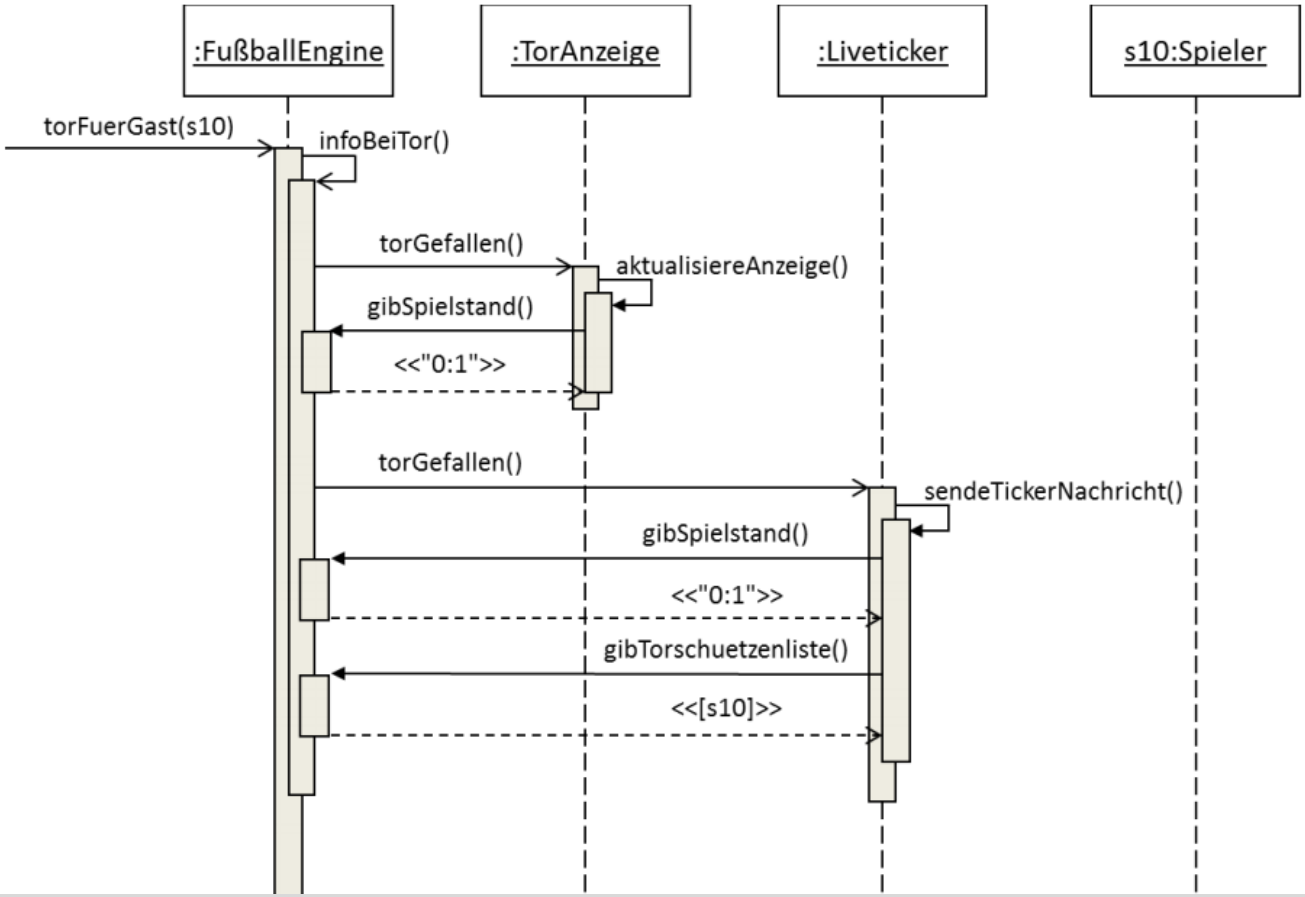
Klausuraufgabe (Hauptklausur SS 2012)



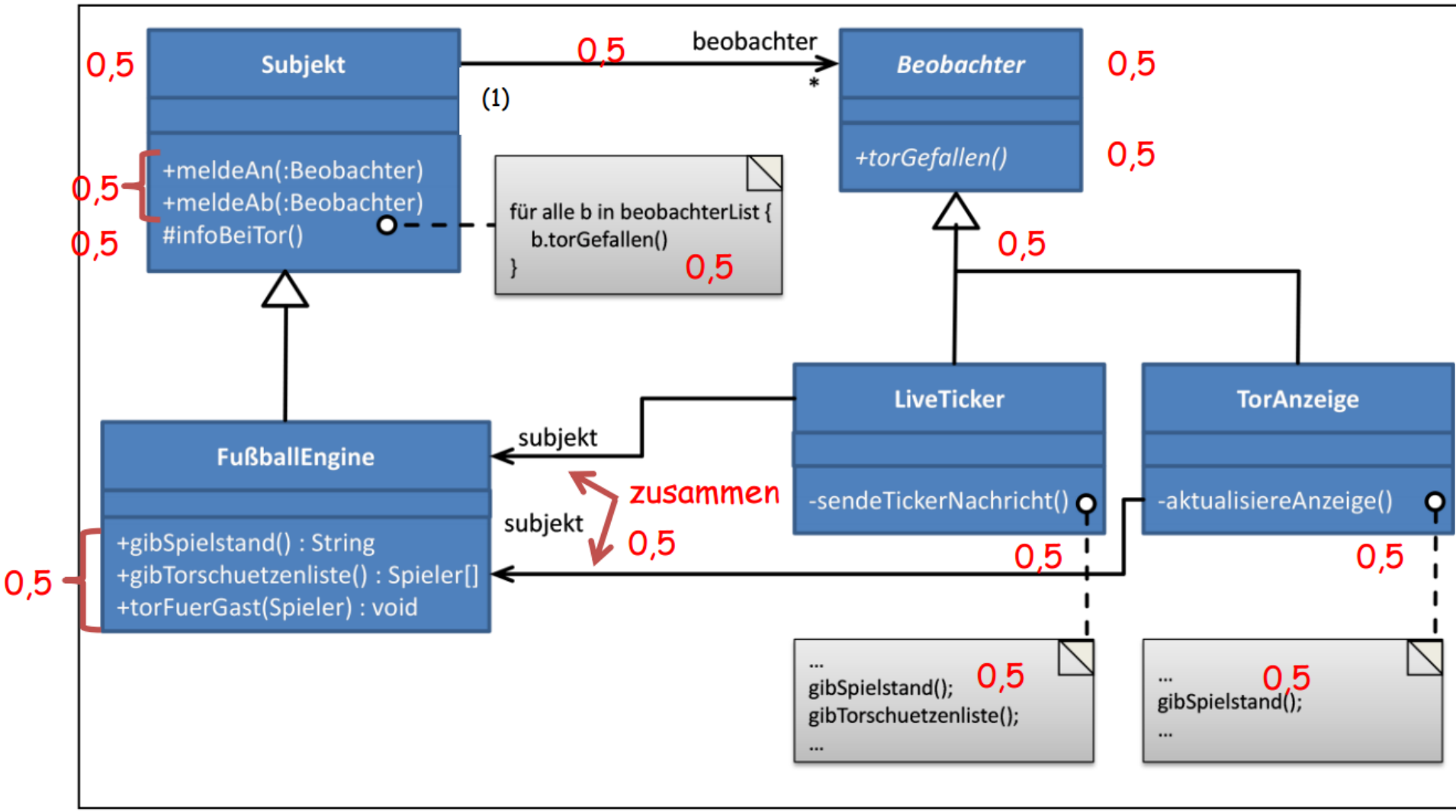
Aufgabe 1

Welches Entwurfsmuster erkennen Sie in diesem Diagramm? Beobachter.

Entwerfen Sie das folgende Klassendiagramm passend zu dem Sequenzdiagramm; es soll alle verwendeten Klassen und Methoden enthalten. Kennzeichnen Sie die Zugreifbarkeiten der Methoden mit den Symbolen +, -, #; seien Sie dabei möglichst restriktiv. Verzichten Sie auf die Modellierung von Attributen. Kennzeichnen Sie die Elemente des Entwurfsmusters und deren Funktion.



Musterlösung



Tipps - 4. Übungsblatt

Aufgabe 1: iMage-GUI

- macht die “kleinen“ Bonusaufgaben
  $\Longrightarrow$ relativ leichte Punkte

Tipps - 4. Übungsblatt

Aufgabe 1: iMage-GUI

- macht die “kleinen“ Bonusaufgaben
  $\Longrightarrow$ relativ leichte Punkte
- schaut euch die verschiedenen LayoutManager aus Java Swing an
  $\Longrightarrow$ verschiedene LayoutManager möglich (via mehrerer Container, z.B. JPanel)

Tipps - 4. Übungsblatt

Aufgabe 1: iMage-GUI

- macht die “kleinen“ Bonusaufgaben
  $\Longrightarrow$ relativ leichte Punkte
- schaut euch die verschiedenen LayoutManager aus Java Swing an
  $\Longrightarrow$ verschiedene LayoutManager möglich (via mehrerer Container, z.B. JPanel)

Aufgabe 2: Zustandsdiagramm

- nochmal Syntax anschauen
  $\Longrightarrow$ Was darf in Zustandsdiagramm, was nicht? (laut VL)
- von Hand!

Tipps - 4. Übungsblatt

Aufgabe 3: Git

- echo ”hallo”>> test.txt schreibt hallo in test.txt
- git-Dokumentation anschauen

Tipps - 4. Übungsblatt

Aufgabe 3: Git

- echo ”hallo”>> test.txt schreibt hallo in test.txt
- git-Dokumentation anschauen

Aufgabe 4: Architekturstile

- Jmjrst (gedanklich) umbauen
- Zusammenhang der Klassen anschauen (z.B. Main-Klasse)

Denkt dran!

Abgabe

- Deadline am 13.6 um 12:00
- A2-4 handschriftlich!

Bis dann! (dann := 19.06.17)