

Softwaretechnik 1 - 4. Tutorium

Tutorium 03

Felix Bachmann | 26.06.2017

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)

- 1 Orga
- 2 Recap
- Stellvertreter
- 4 Vermittler
- Gruppenarbeit
- 6 Memento
- Befehl
- 8 Tipps

Felix Bachmann - SWT1

Allgemeines



Ansage der Übungsleiter

ab jetzt keine Abgabe per Mail mehr!

⇒ auch nicht in Ausnahmefällen

Allgemeines



Ansage der Übungsleiter

- ab jetzt keine Abgabe per Mail mehr!
 - ⇒ auch nicht in Ausnahmefällen
- keine Abgabe von semi-handschriftlichen Ausarbeitungen mehr!

Felix Bachmann - SWT1

4. Übungsblatt Statistik



| Orga | Recap |
|-----------------------|-----------|
| 00000 | 000000000 |
| Felix Bachmann - SWT1 | |



Allgemein





Aufgabe 1 (Zustandsdiagramm - LEZ): ∅





Aufgabe 2 (Abbottsche Methode): Ø

Aufgabe 3 (iMage-GUI): ∅





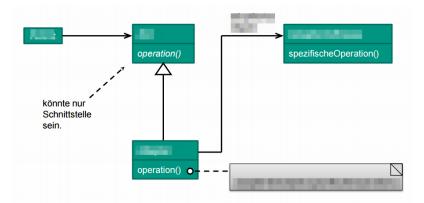
Aufgabe 4 (Geheimnisprinzip): ∅



haben uns erste Entkopplungmuster angeschaut



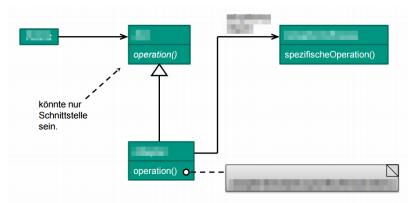
- haben uns erste Entkopplungmuster angeschaut
 - ⇒ Beobachter, Iterator, Adapter



Welches Entwurfsmuster?



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



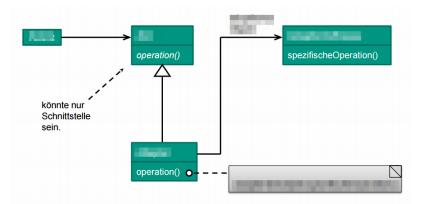
Welches Entwurfsmuster? (Objekt-)Adapter

| Orga | ricoup |
|--------------|-----------|
| 000000 | •00000000 |
| Felix Bachma | nn – SWT1 |

Recan



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

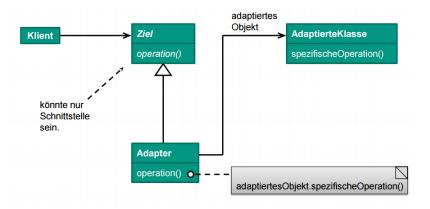


Welche Klassen? Recan

| Orga | riccup |
|--------------|-----------|
| 000000 | 00000000 |
| Felix Bachma | nn – SWT1 |



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





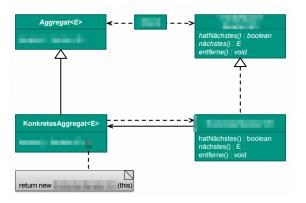
- haben uns erste Entkopplungmuster angeschaut
 - ⇒ Beobachter, Iterator, Adapter



Welches Entwurfsmuster?



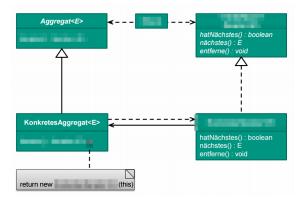
- haben uns erste Entkopplungmuster angeschaut
 - ⇒ Beobachter, Iterator, Adapter



Welches Entwurfsmuster? Iterator



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

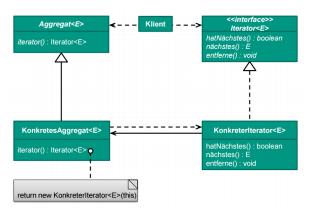


Welche Klassen und Methoden?



haben uns erste Entkopplungmuster angeschaut

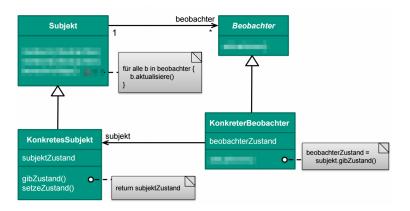
⇒ Beobachter, Iterator, Adapter



14/45

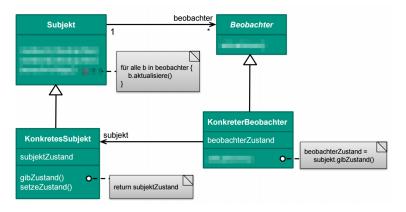


- haben uns erste Entkopplungmuster angeschaut
 - Beobachter, Iterator, Adapter





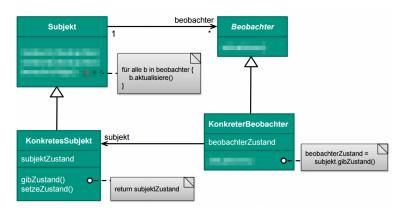
- haben uns erste Entkopplungmuster angeschaut
 - ⇒ Beobachter, Iterator, Adapter



Ist wohl ein Beobachter:)



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



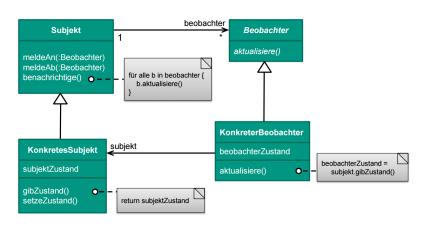
Ist wohl ein Beobachter :) Klassen, Methoden?

| Orga | Recap | |
|-----------------------|--------|--|
| 000000 | 000000 | |
| Felix Bachmann - SWT1 | | |

26.06.2017



haben uns erste Entkopplungmuster angeschaut
 ⇒ Beobachter, Iterator, Adapter



Vermittler

26.06.2017

Kategorien der Entwurfsmuster



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

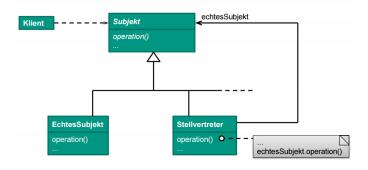
Felix Bachmann - SWT1

Tipps

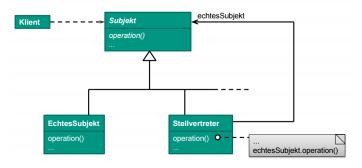


Problem

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





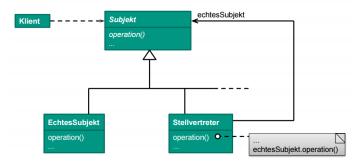


Entkopplung?

Recap

Stellvertreter



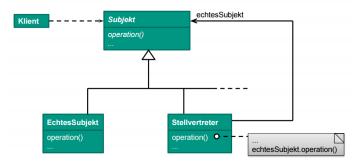


Entkopplung?

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

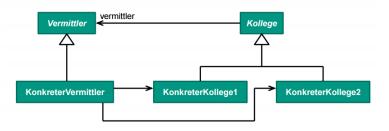
mehrere voneinander abhängige Objekte

⇒ Zustände der Objekte von anderen Zuständen abhängig

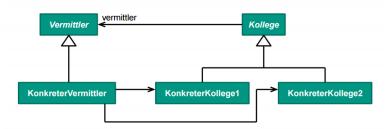


Problem

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







Entkopplung?

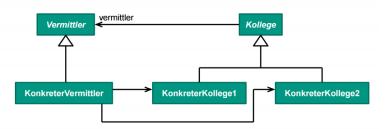




Entkopplung?

Kollegen kennen sich nicht direkt





Entkopplung?

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

Recap

Stellvertreter

Vermittler

Gruppenarbeit

Memento

Befehl

Kategorien der Entwurfsmuster



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

22/45

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

Vermittler

⇒ keine Wiederholung desselben Codes

Varianten-Muster



Übergeordnetes Ziel

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

Jetzt: Gruppenarbeit



- ihr kriegt pro Reihe eine Aufgabe
- ihr habt Zeit zum Bearbeiten
- ihr stellt den anderen eure Lösung vor

Vorstellung Dekorierer Dekorierer hat **Dekorierer: Struktur** Instanzvariable mit Referenz auf eine Komponente Komponente methodeEins() Komponente, methodeZwei() die erweitert werden soll Dekorierer implementieren dieselbe KonkreteKomponente Dekorierer Schnittstelle methodeEins() methodeEins() wie die methodeZwei() methodeZwei() KonkreteKomponente **KonkreterDekoriererA** KonkreterDekoriererB methodeEins() methodeEins() methodeZwei() methodeZwei() neueMethodeA() neueMethodeB() Stellvertreter Vermittler Gruppenarbeit Memento Befehl Tipps Recap

Felix Bachmann - SWT1

0000000000

MuLö Dekorierer



Wo Gemeinsamkeiten?

Die beiden Methoden methodeEins() und methodeZwei().

Wo Variation?

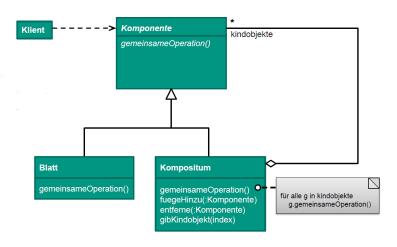
In den KonkretenDekorierern bzw. ihren Methoden. Hier: neueMethodeA(), neueMethodeB().

Wozu Instanzvariable?

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

Vorstellung Kompositum





MuLö Kompositum



Wo Gemeinsamkeiten?

gemeinsameOperation().

Wo Variation?

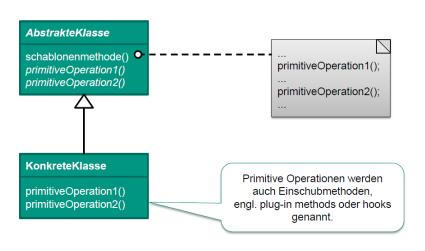
In Blatt/Kompositum-Klassen mit verschiedenen zusätzlichen Operationen.

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

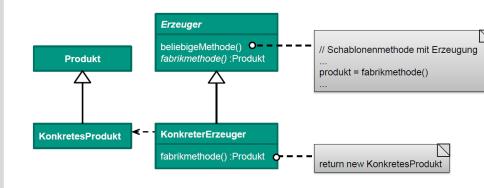
Einschubmethode ist eine der Methoden, die von der Schablonenmethode aufgerufen wird und deren Implementierung in den Unterklassen stattfindet.

Vermittler

26.06.2017

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.

Recap

26.06.2017

Kategorien der Entwurfsmuster



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

Felix Bachmann - SWT1

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)

Vermittler

aber unabhängig von dem Zweck des Objekts!



Problem

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



Problem

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

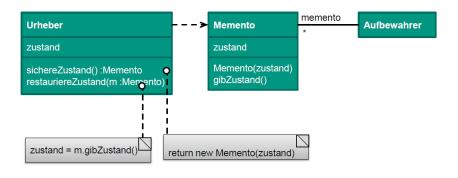
Vermittler

ohne Kapselung zu verletzten!

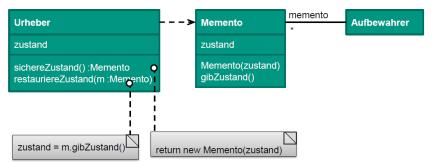


Problem

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







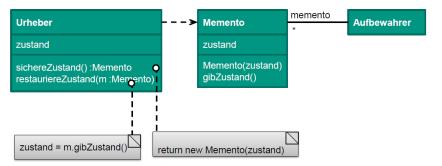
Problem gelöst?

ooooo ooooo Felix Bachmann – SWT1

Recap 000000000 Stellvertreter 00 Vermittler 00 Gruppenarbeit 0000000000 Memento ○○○● Befehl 0000

Tipps 00000





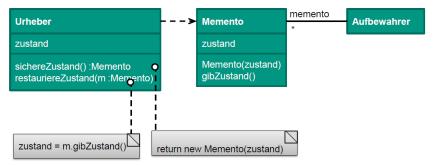
Problem gelöst?

Ja









Problem gelöst?

Ja

Zustand durch Memento externalisiert

Felix Bachmann - SWT1

Recap

Stellvertreter

Vermittler

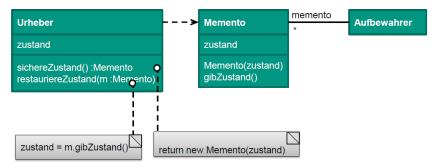
Gruppenarbeit

Memento 0000

Befehl

Tipps





Problem gelöst?

Recap

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

Kategorien der Entwurfsmuster



- Entkopplungs-Muster fertig
- Varianten-Muster fertig
- Zustandshandhabungs-Muster fertig
- Steuerungs-Muster
 - Befehl
 - (master/worker)
- Bequemlichkeits-Muster

Steuerungs-Muster



Übergeordnetes Ziel

steuern den Kontrollfluss

Steuerungs-Muster



Übergeordnetes Ziel

steuern den Kontrollfluss

⇒ zur richtigen Zeit richtige Methoden aufrufen



Problem

Parametrisieren von Objekten mit einer auszuführenden Aktion

Vermittler

 Orga
 Recap

 00000
 0000000

 Felix Bachmann - SWT1



Problem

- Parametrisieren von Objekten mit einer auszuführenden Aktion
- komplexe Operationen aus primitiven Operationen aufbauen



Problem

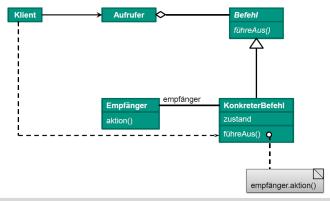
- Parametrisieren von Objekten mit einer auszuführenden Aktion
- komplexe Operationen aus primitiven Operationen aufbauen
- Ausführung in Warteschlange

Felix Bachmann - SWT1

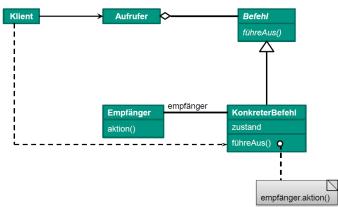


Problem

- Parametrisieren von Objekten mit einer auszuführenden Aktion
- komplexe Operationen aus primitiven Operationen aufbauen
- Ausführung in Warteschlange







Was haben wir erreicht?

 Austauschbarkeit: Befehle unabhängig vom Aufrufer, universell einsetzbar

Recap

Stellvertreter

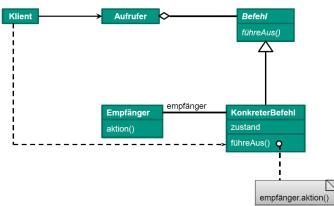
Vermittler

Gruppenarbeit

Memento

Befehl 0000





Was haben wir erreicht?

- Austauschbarkeit: Befehle unabhängig vom Aufrufer, universell einsetzbar
- Logik komplett im Befehl ⇒ keine Inkonsistenzen bei Benutzung

26.06.2017





Aufgabe 1: Manager-Deutsch und Architekturstile

Architekturstile nochmal anschauen



Aufgabe 1: Manager-Deutsch und Architekturstile

Architekturstile nochmal anschauen

Aufgabe 2: Iterator für Plug-Ins

Iterator-Muster selbst benutzen

Vermittler

26.06.2017



Aufgabe 3: Geometrify mit Entwurfsmustern

überlegen, welches Entwurfsmuster warum Sinn macht



Aufgabe 3: Geometrify mit Entwurfsmustern

■ überlegen, welches Entwurfsmuster warum Sinn macht

Vermittler

Aufgabe 4: Geometrify umstrukturieren

■ Überlegungen aus Aufgabe 3 umsetzen



Aufgabe 5: GUI erweitern

■ nochmal ServiceLoader ⇒ diesmal mit Primitiven

Denkt dran!



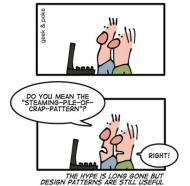
Abgabe

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

Bis dann! (dann := 10.07.17)









Recap Felix Bachmann - SWT1