

Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

Softwaretechnik 1 Tutorium

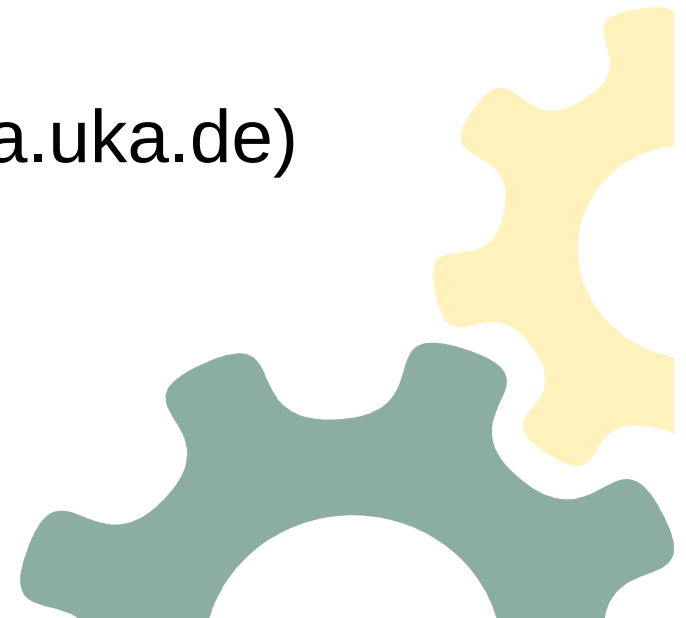
08. Juni 2009

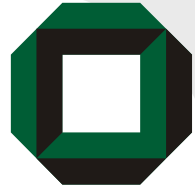
Matthias Thoma (s_thoma@ira.uka.de)



Fakultät für **Informatik**

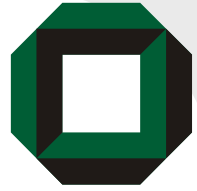
Lehrstuhl für Programmiersysteme





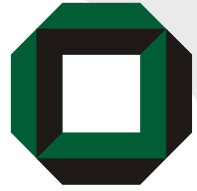
Heute

- Übungsblatt Nr. 3
- Entwurfsmuster
 - Strategie
 - Besucher
 - Adapter
 - Fassade
 - Dekorierer
- Hinweise zum nächsten Übungsblatt



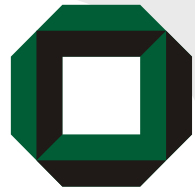
Kommentare zum letzten Übungsblatt

- Ziemlich gut ausgefallen



Entwurfsmuster - kurzgefasst

- Ein Software-Entwurfsmuster beschreibt eine Familie von Lösungen für ein Software-Entwurfsproblem.
- Ziel: Wiederverwendbarkeit von Entwurfswissen.
- Bestehend aus:
 - Problem
 - Lösung
 - Kontext



Buchempfehlungen

- **Head First Design Patterns**

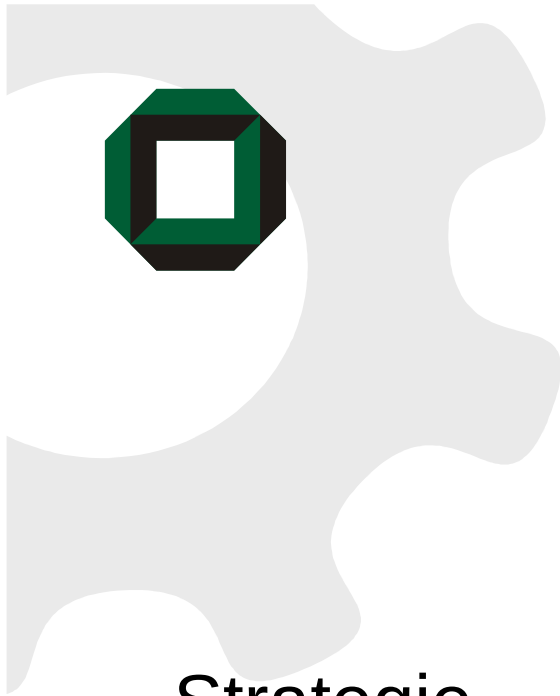
Eric Freeman & Elisabeth Freeman,
with Kathy Sierra & Bert Bates



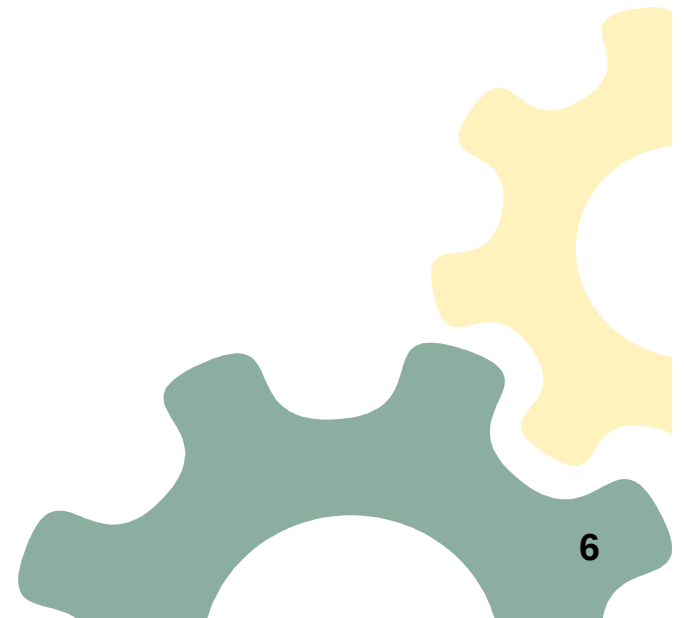
- **Pattern-orientierte Softwarearchitektur**

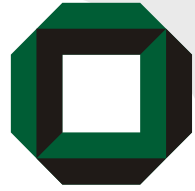
Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal





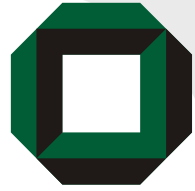
Strategie





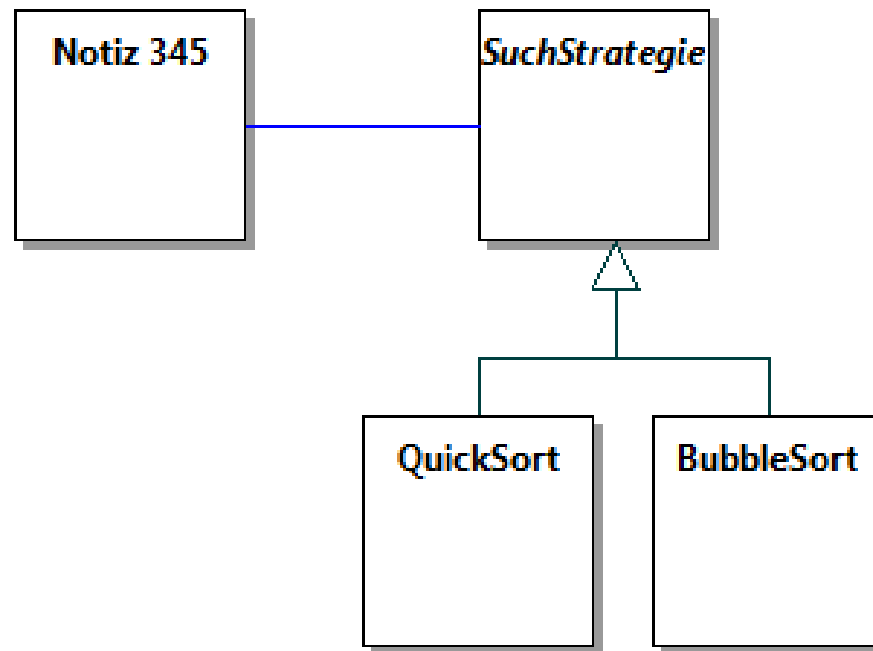
Strategie

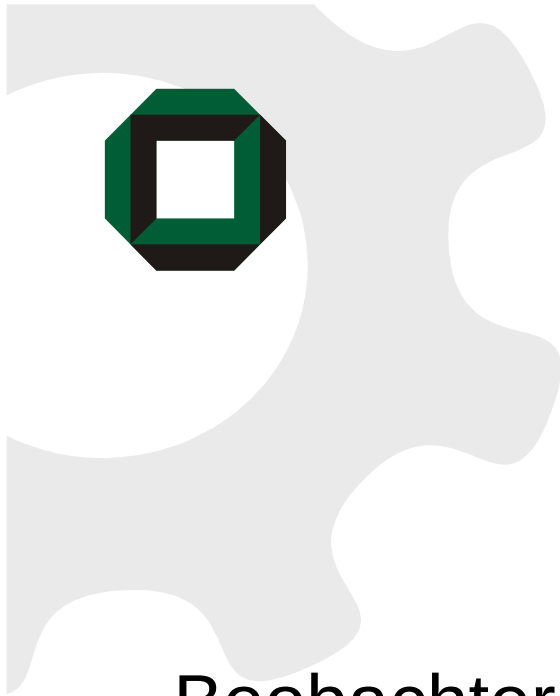
- Zweck: Eine Familie von Algorithmen soll gekapselt werden, mit der Möglichkeit, sie beliebig auszutauschen.



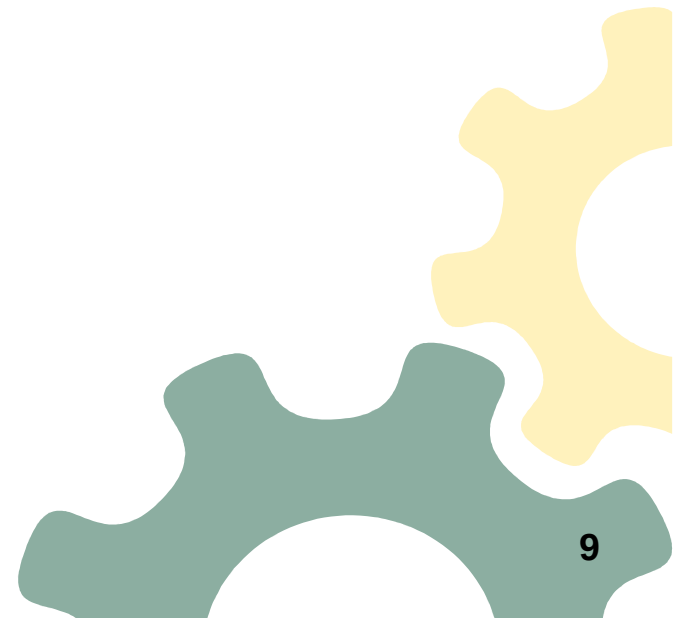
Strategie - Beispiel

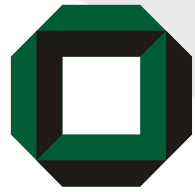
- Die Applikation „Notiz 3-4-5“ soll zwischen QuickSort und BubbleSort wählen können.



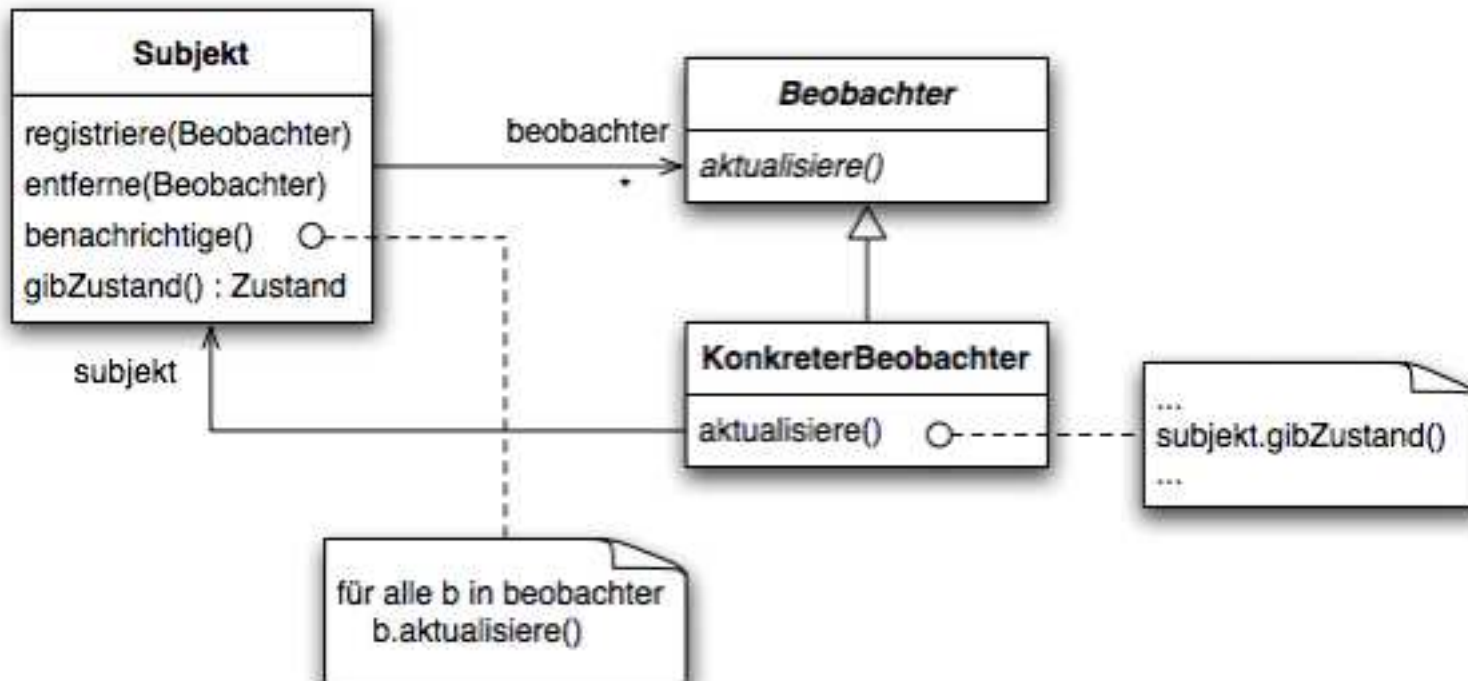


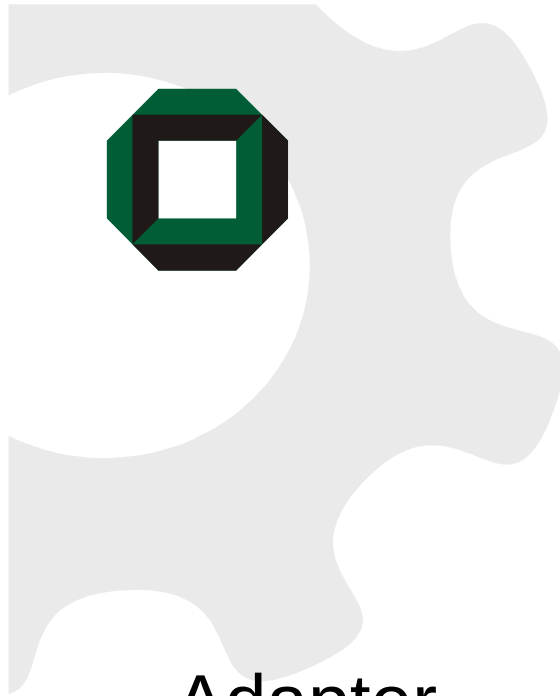
Beobachter



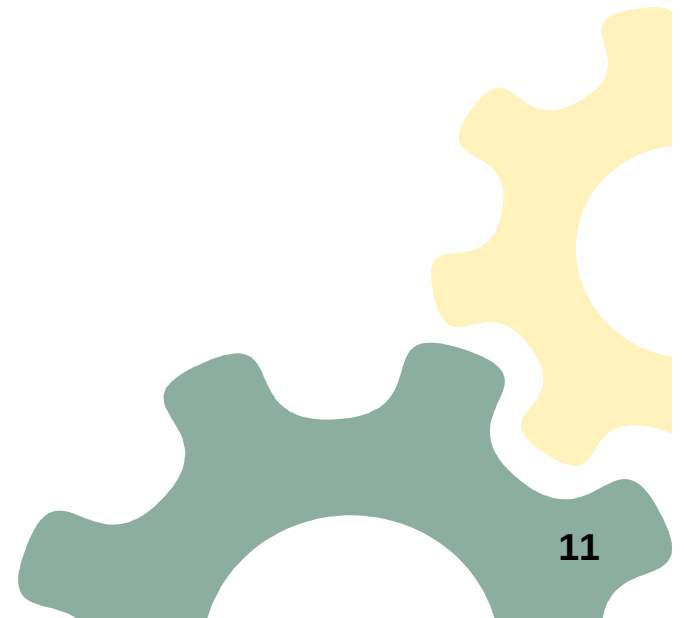


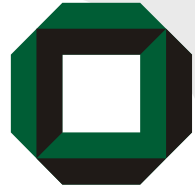
Beobachter





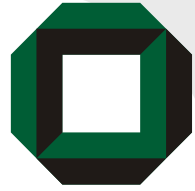
Adapter





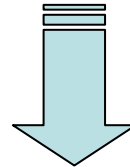
Adapter

- Problem: Zwei Klassen/Komponenten mit unterschiedlichen inkompatiblen Schnittstellen sollen zusammenarbeiten.
- Implementierungsalternativen:
 - Über Delegation (Objektadapter)
 - Über Vererbung (Klassenadapter)

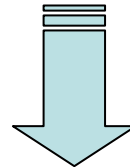


Adapter - Ablauf

Der Klient führt Anfrage auf dem
Adapterobjekt aus



Adapter ruft Operation der
adaptierten Klasse auf



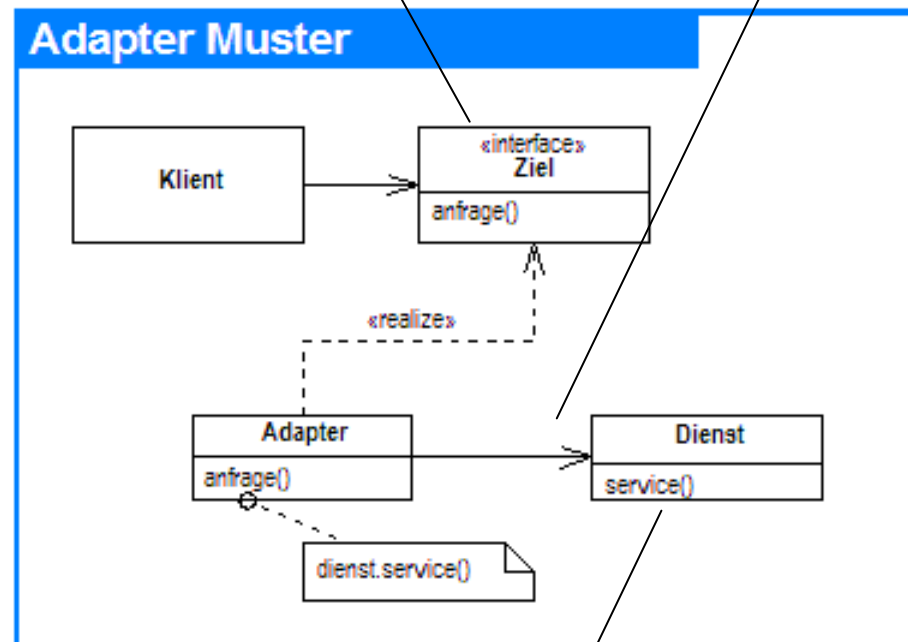
Anfrage wird beantwortet



Adapter - Delegation

Schnittstelle die Klient erwartet.

Delegation



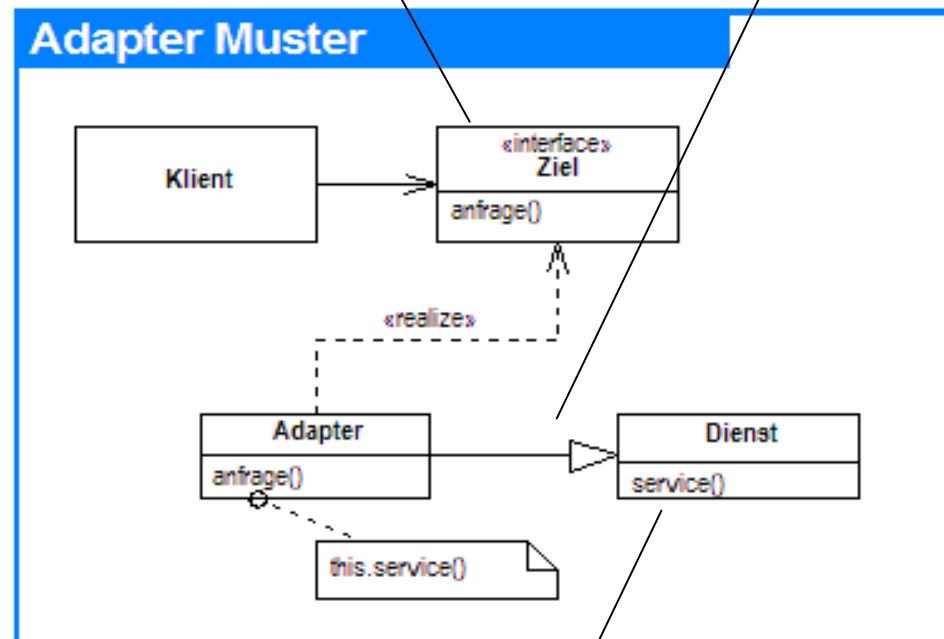
Dienst mit inkompatibler Schnittstelle



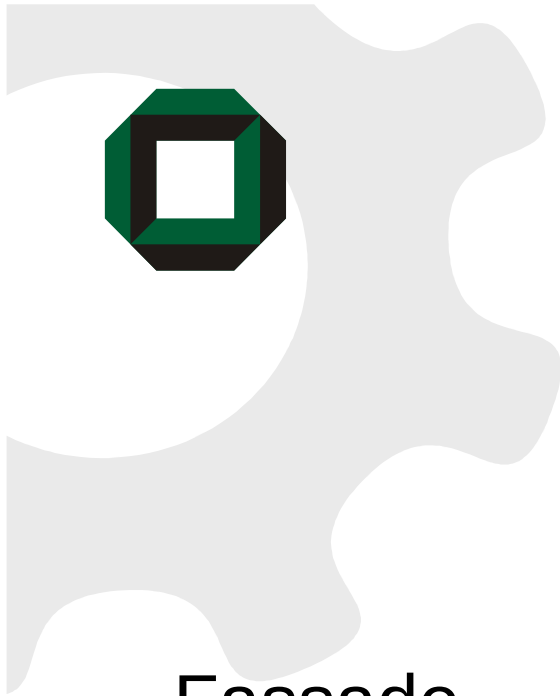
Adapter - Delegation

Schnittstelle die Klient erwartet.

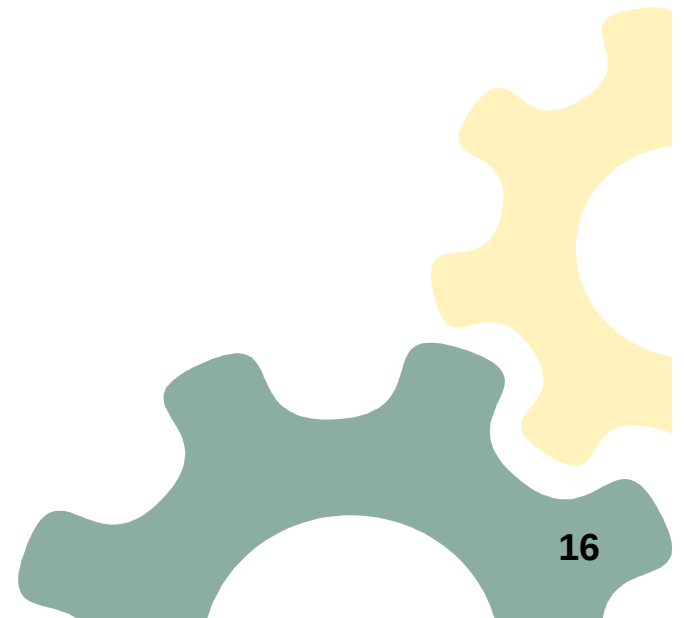
Vererbung

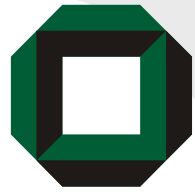


Dienst mit inkompatibler Schnittstelle



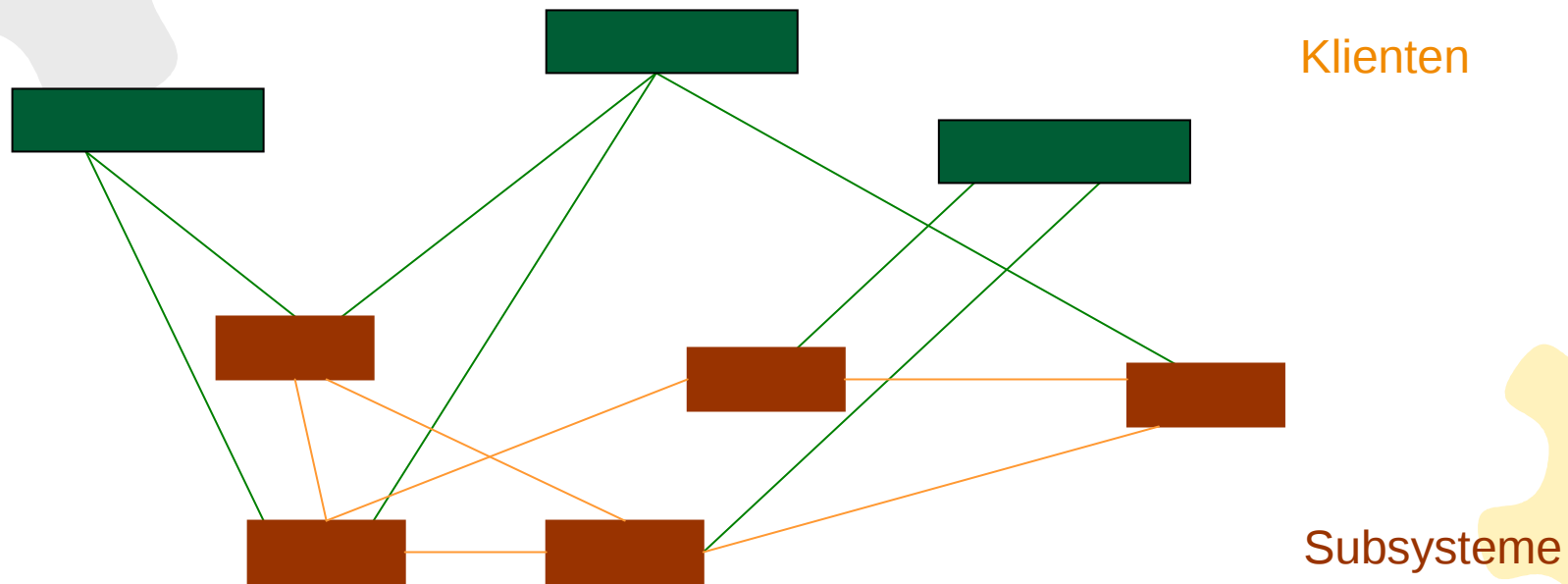
Fassade

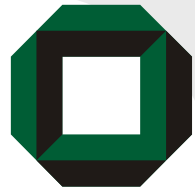




Fassade

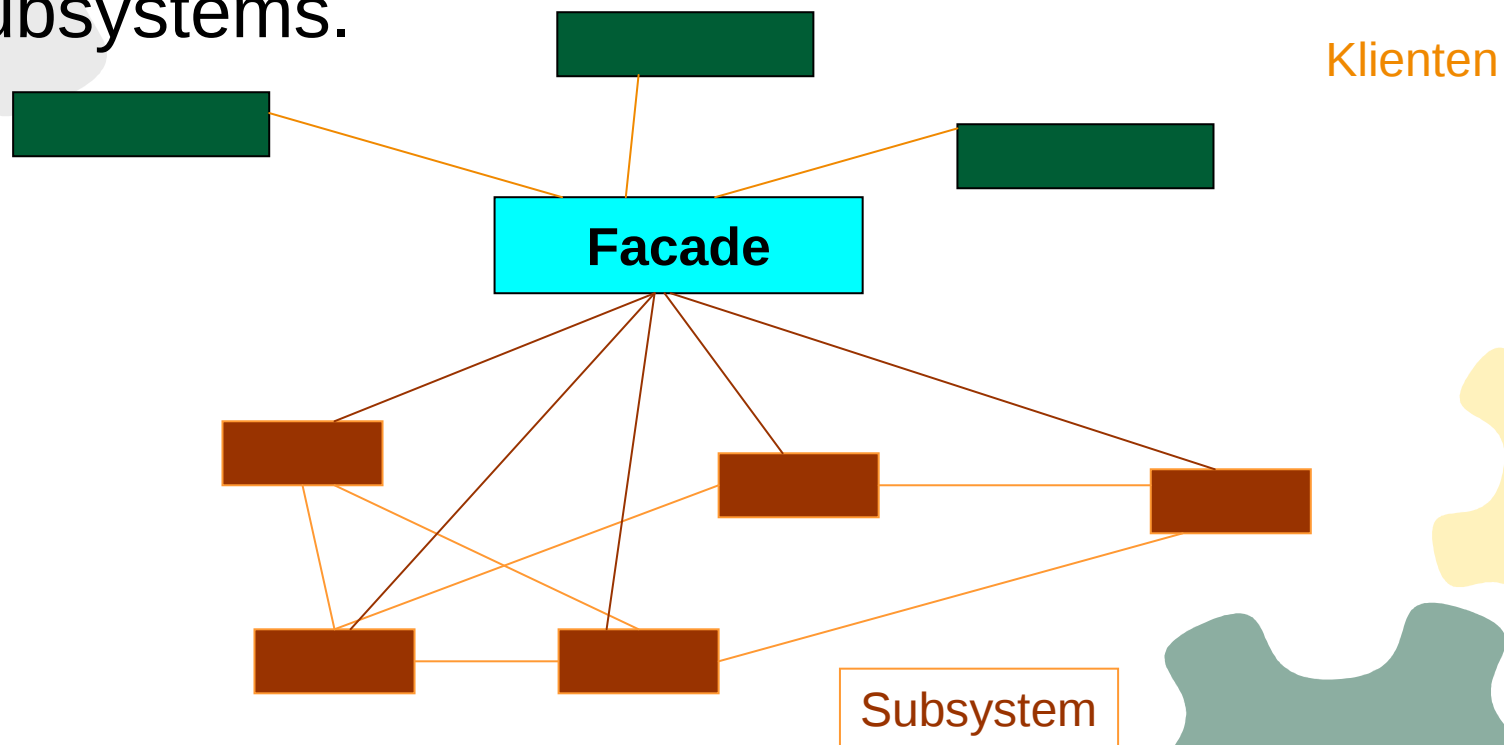
- Problem: Klienten müssen mit unterschiedlichen Subsystem arbeiten.

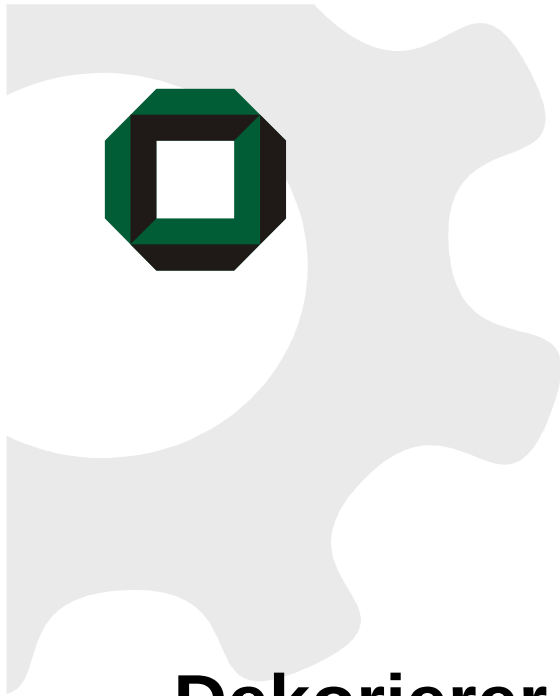




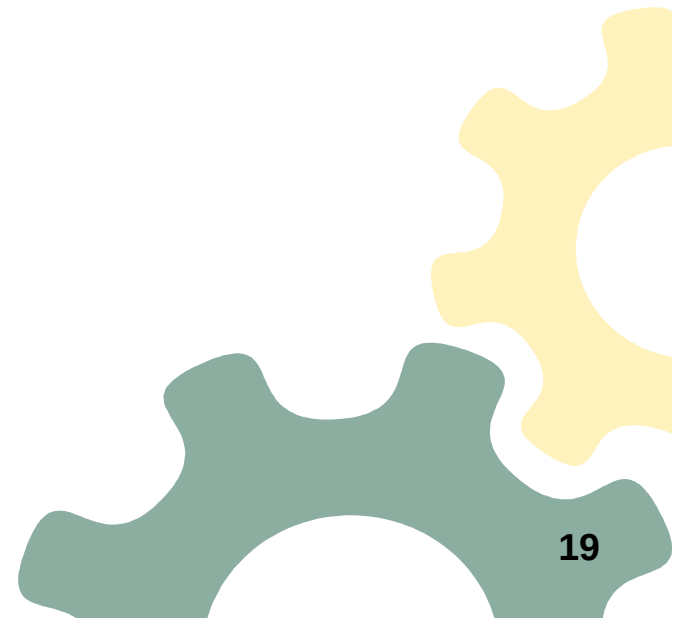
Fassade - Lösung

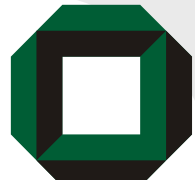
Lösung: Biete eine einheitliche Schnittstelle zu einer Menge von Schnittstellen eines Subsystems.



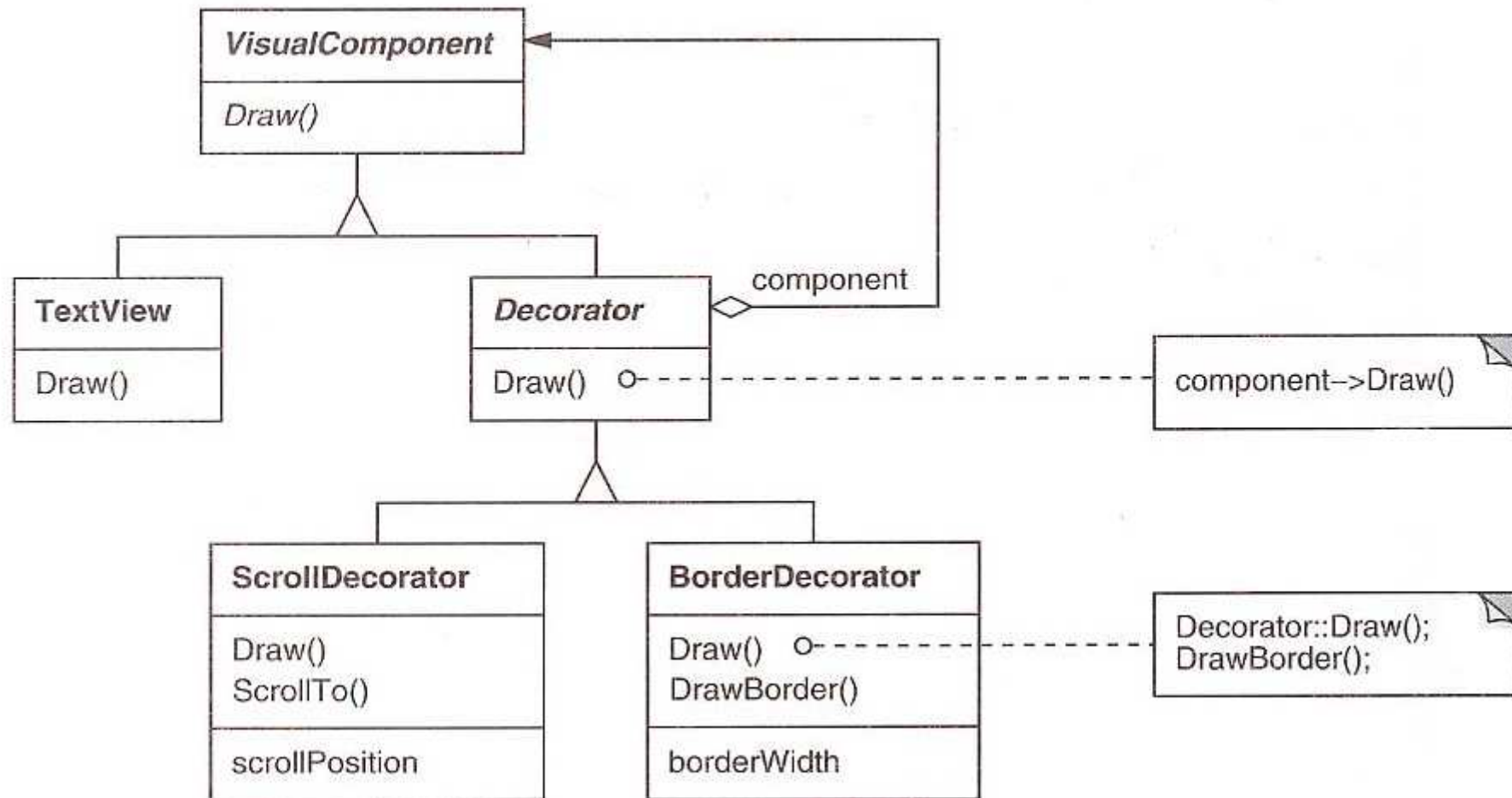


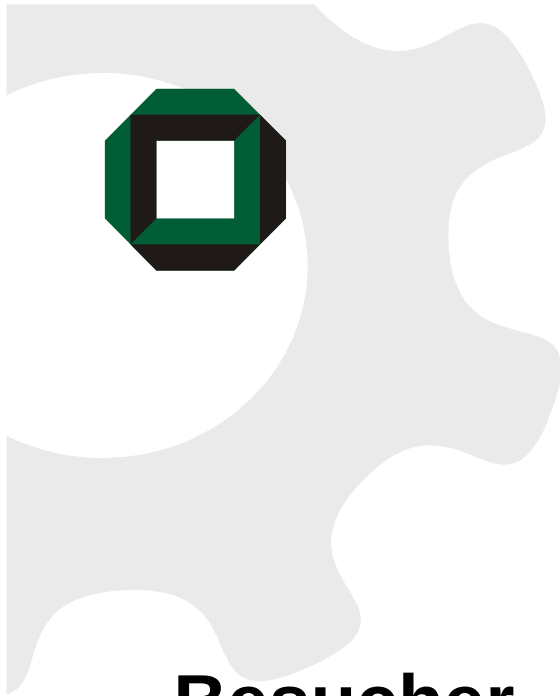
Dekorierer



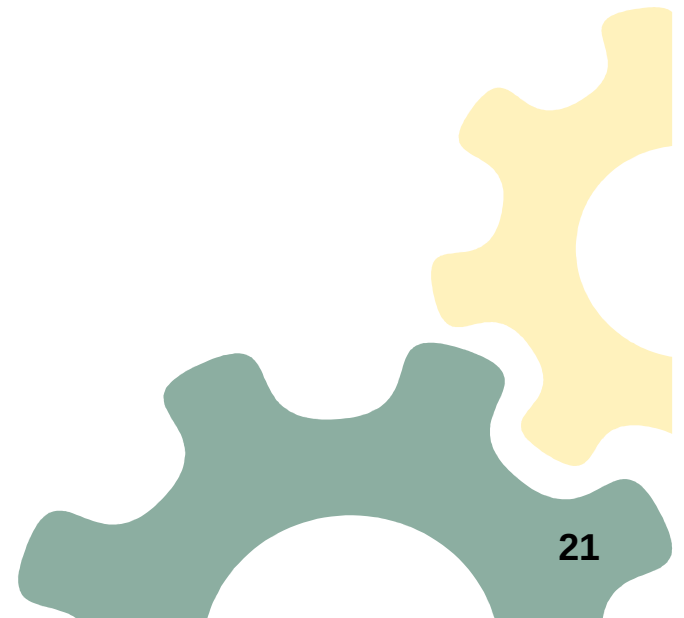


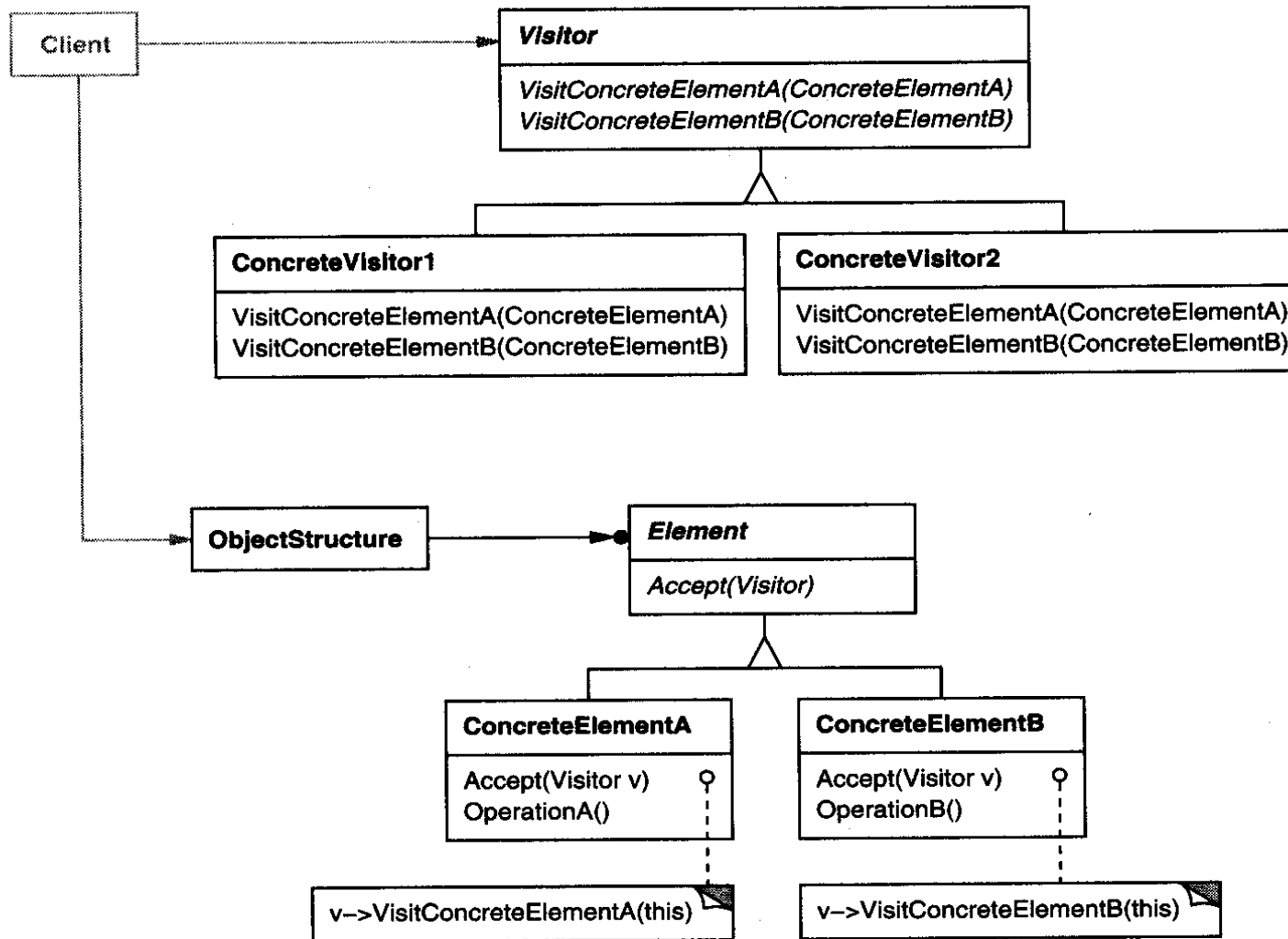
Dekorierer

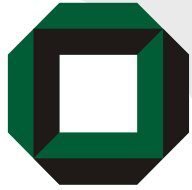




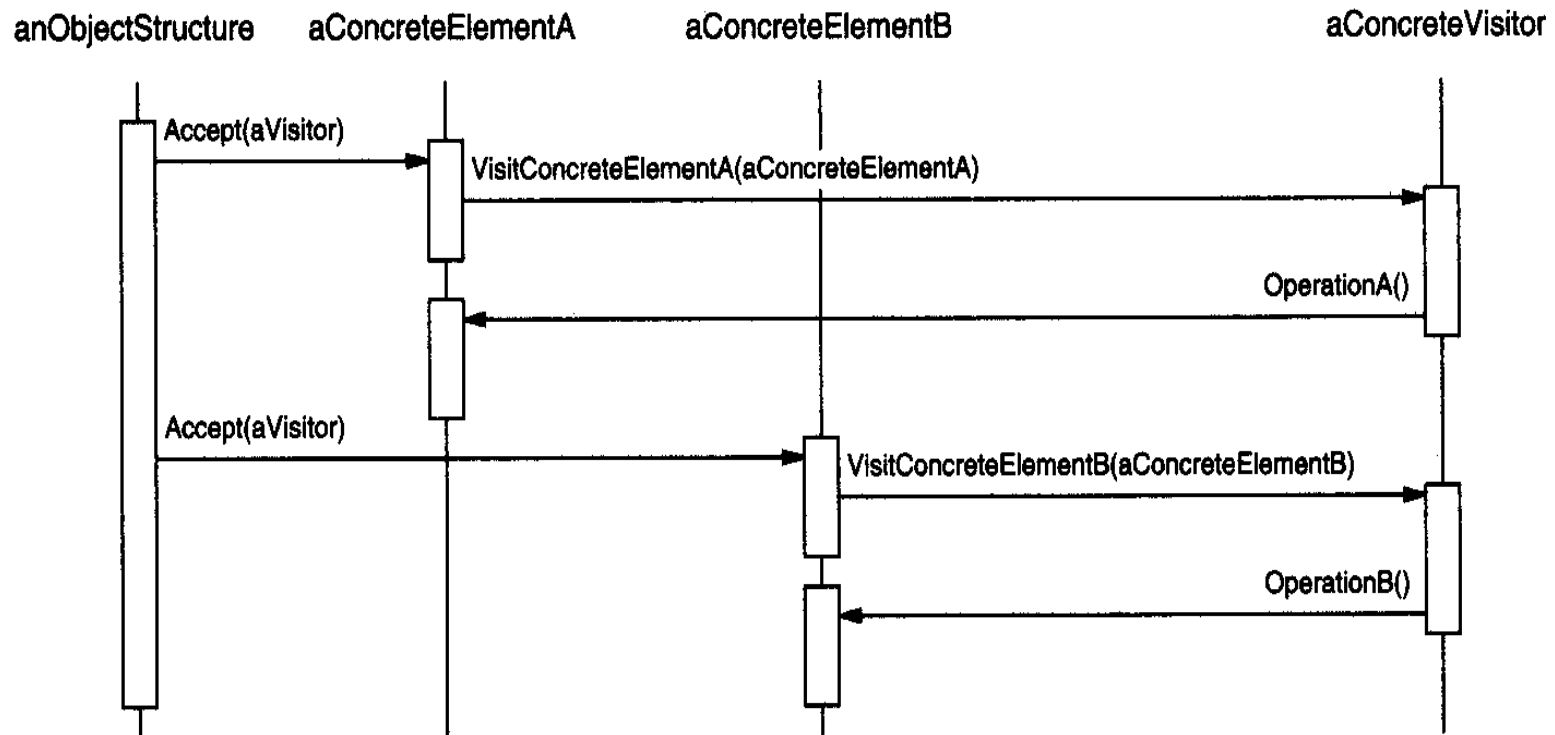
Besucher

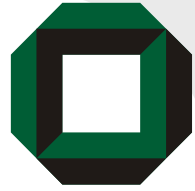






Besucher





Parallelität in JAVA

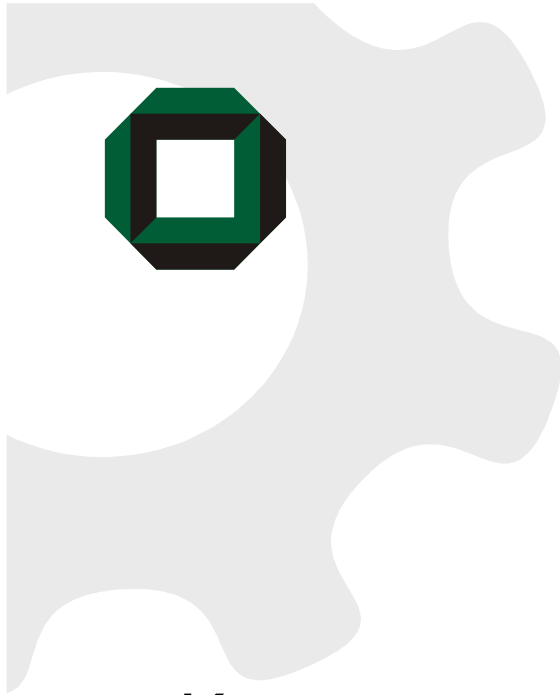
- Folien

<http://www.ipd.uka.karlsruhe.de/Tichy/uploads/folien/150/Cluster03Java.pdf>

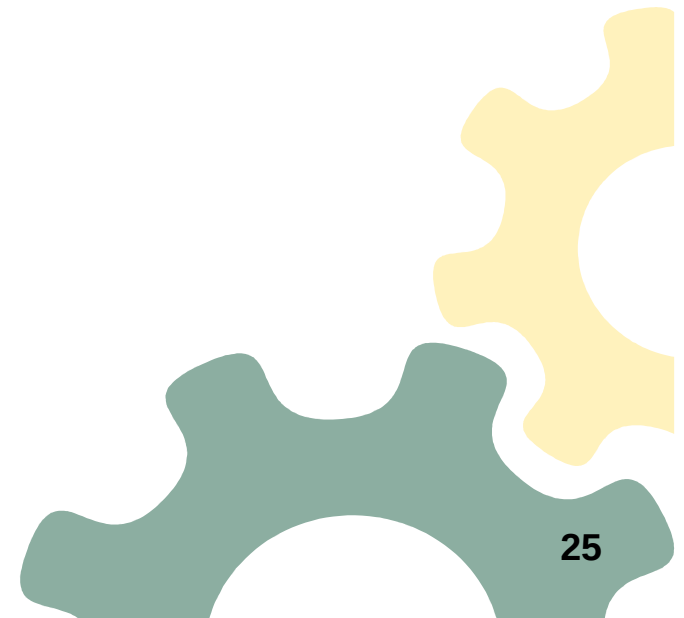
<http://www.ipd.uka.de/Tichy/uploads/folien/157/SEPP05-ParallelitaetJava.pdf>

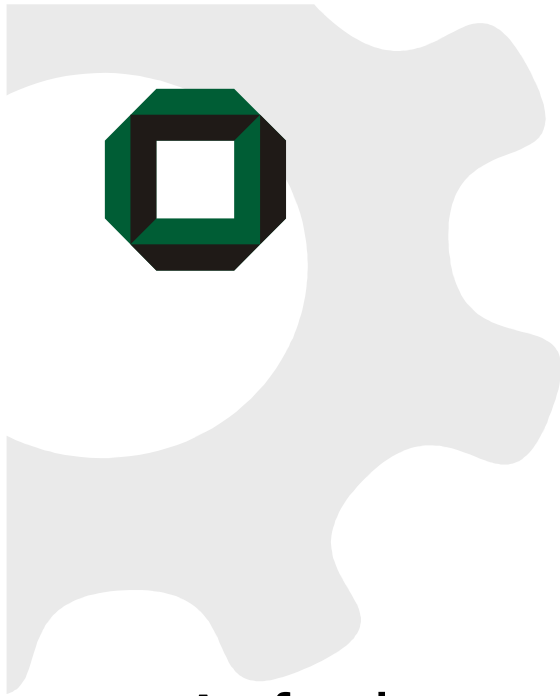
Zu volatile:

<http://www.ipd.uka.de/Tichy/uploads/folien/150/Cluster04JavaSpeichermodell.pdf>

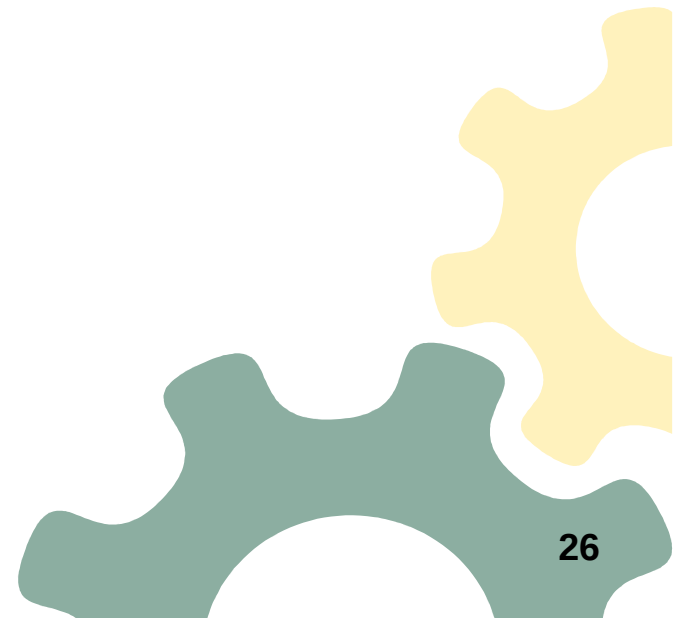


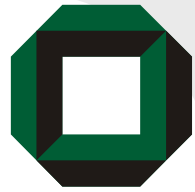
Kommentare zum nächsten Übungsblatt



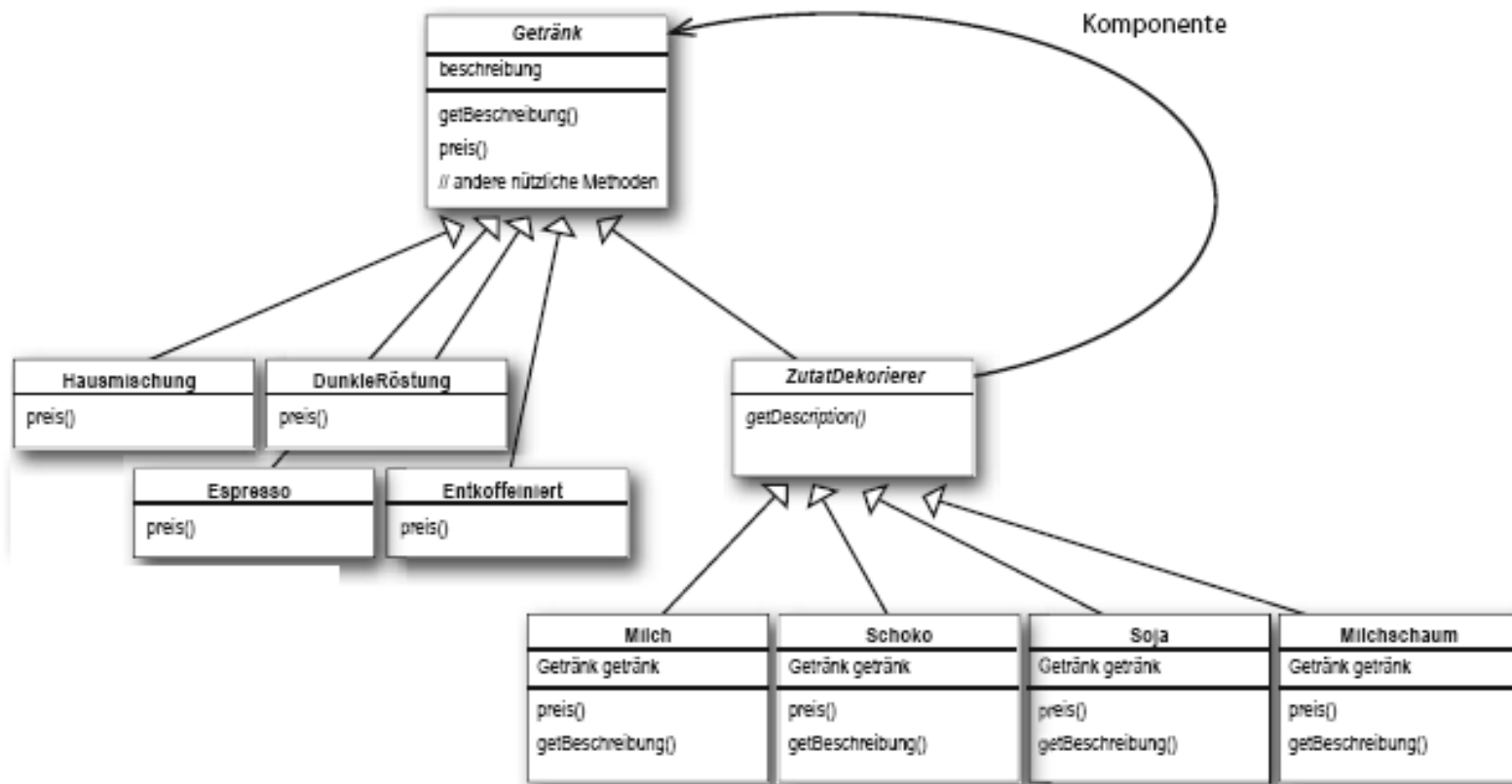


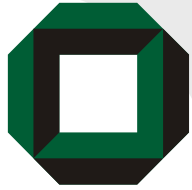
Aufgaben





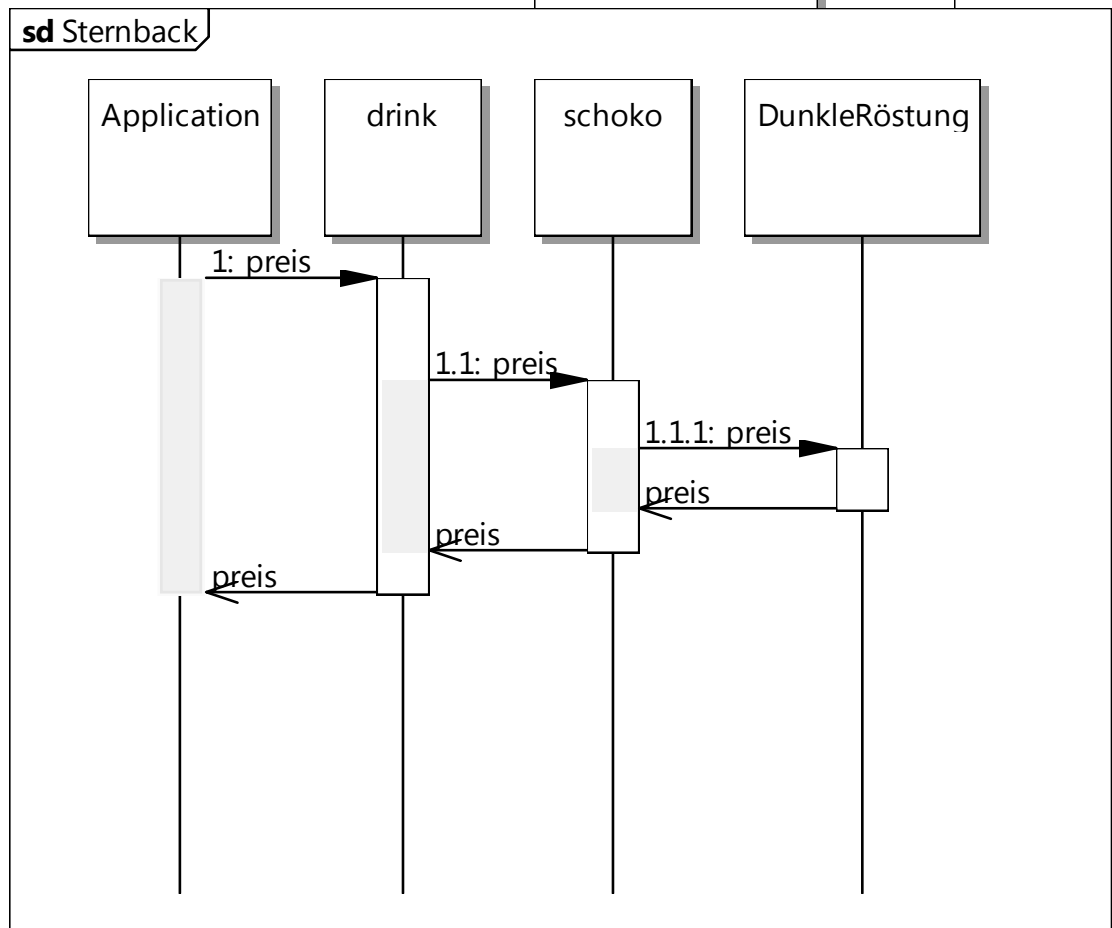
Aufgabe 1 - Klassendiagramm

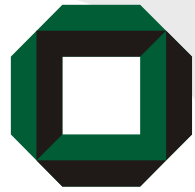




Aufgabe 1 - Sequenzdiagramm

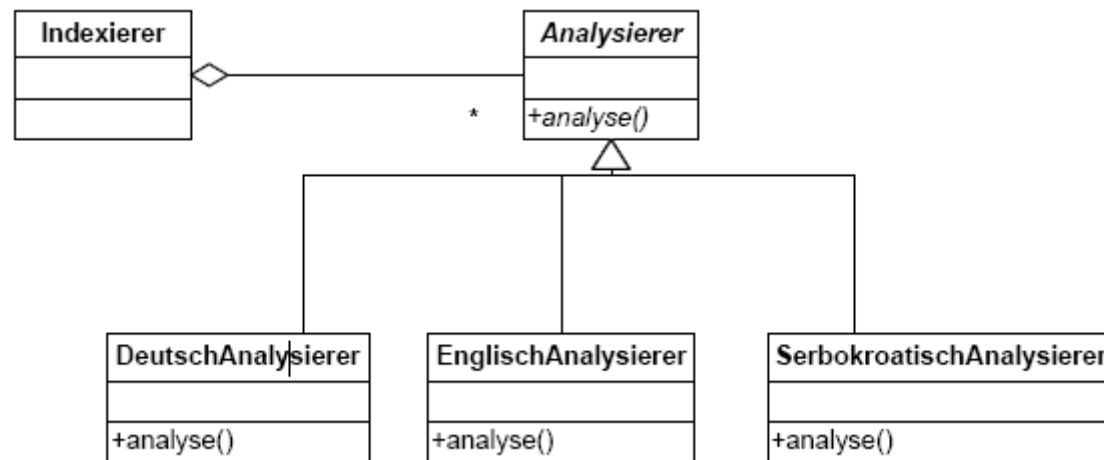
```
Getränk drink = new DunkleRöstung();  
drink = new Schoko(drink);  
drink = new Milchschaum(dringt);  
System.out.println(  
    drink.getBeschreibung() + " " +  
    drink.preis() + " €");
```

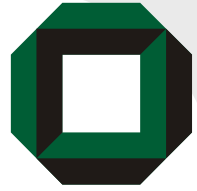




Aufgabe 3.1

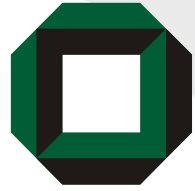
- Strategie





Aufgabe 3.2 / 3.3 / 3.4

- 3.2: Adapter um die Texte in den Formaten HTML bzw. PDF auf die Document Schnittstelle abzubilden
- 3.3: Fabrik
- 3.4: Stellvertreter



Aufgabe 4

1. Richtig
2. Richtig
3. Richtig
4. Falsch
5. Richtig
6. Richtig
7. Richtig
8. Richtig
9. Falsch
10. Richtig