



# Softwaretechnik 2015/2016

PST Lehrstuhl

Prof. Dr. Matthias Hölzl

Joschka Rinke



- Übung 11:**
- 14.01.2016**
- **Schon zur Klausur angemeldet?**
  - **Fragen**
  - **Reflection in LISP**
  - **Besprechung Blatt10**

# Aufgabe 1 – Objektorientierte Analyse

**OOA und OOD sind mögliche Bestandteile des SW-Entwicklungsprozesses**

**Objektorientierte Analyse (Anforderungen erfassen und beschreiben):  
Anforderungsfall-Analyse (Use Case-Analyse) und Entwicklung eines**

- 1) statischen Teilmodells**
  - 2) dynamischen Teilmodells**
- iterativer Prozess**

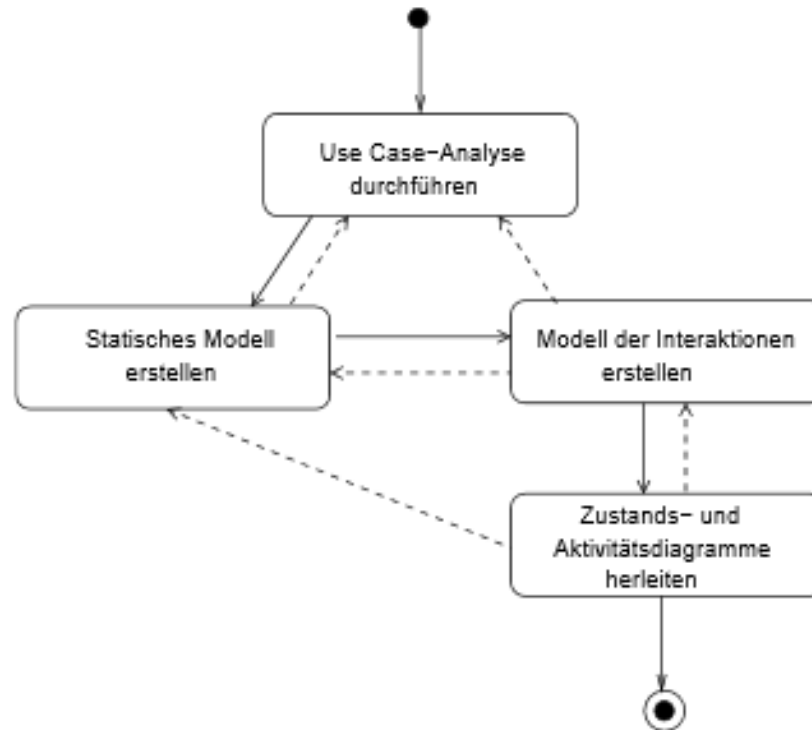
**Objektorientiertes Design (Systementwurf erstellen):**

**→ Modell der konkreten Systemimplementierung (Systementwurf)**

# Aufgabe 1 – Objektorientierte Analyse

**Objektorientierte Analyse (Vorgehensweise):**

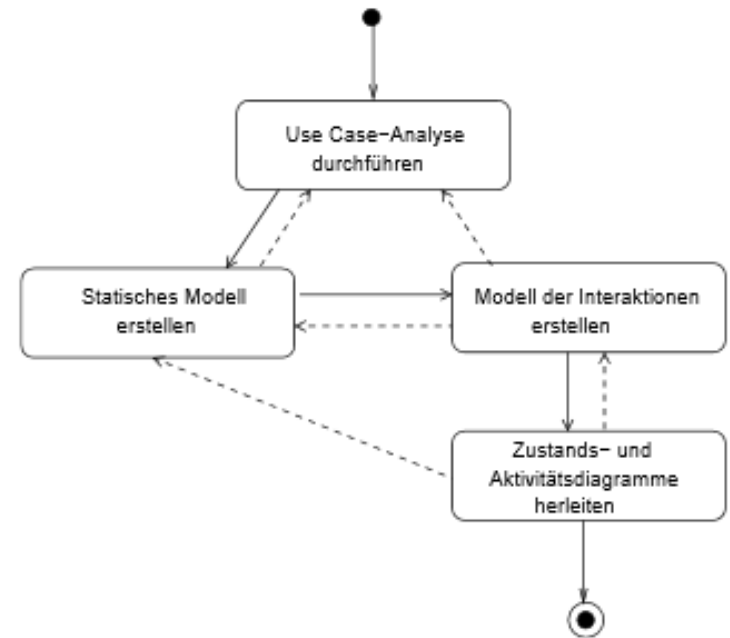
**Use Cases → statisches Modell → dynamisches Modell → Diagramme erstellen**





**Was für ein Diagrammtyp liegt hier vor?**

- Sequenzdiagramm
- Klassendiagramm
- Kommunikationsdiagramm
- Aktivitätsdiagramm
- Zustandsdiagramm



# Aufgabe 1 – Objektorientierte Analyse

**Statisches Teilmodell:**

**Beziehungen zwischen Klassen → Klassendiagramme**

**Dynamisches Teilmodell:**

**Operationen und Kommunikation zwischen Klassen**

**→ Interaktionsdiagramme (Sequenz-/ Kommunikationsdiagramme)**

**→ daraus werden dann Aktivitäts- und Zustandsdiagramme erstellt**

# Aufgabe 1 – Objektorientierte Analyse

## **OOA:**

**Entwickler analysieren die Problemstellung;**

**Ziel ist es Anforderungen zu erfassen und Systemeigenschaften zu modellieren, dazu:**

- Anwendungsfälle dokumentieren**
- statisches Teilmodell entwerfen**
- dynamisches Teilmodell entwerfen**
- Zustands-/ Aktivitätsdiagramme herleiten (wenn notwendig)**
- evtl. Pflichtenheft erstellen**
- Ergebnis der OOA sollte vollständige Beschreibung d. Systemverhaltens sein**

## **Agile SW:**

**Beteiligte Rollen erarbeiten gemeinsam User Stories**

- Tasks extrahieren**
- Product Backlog erstellen**
- User Stories können immer wieder hinzugefügt werden**

# Aufgabe 1 – Objektorientierte Analyse

**Anwendungsfälle (Use Cases) dokumentieren, indem ein Use Case-Modell entworfen wird, dazu:**

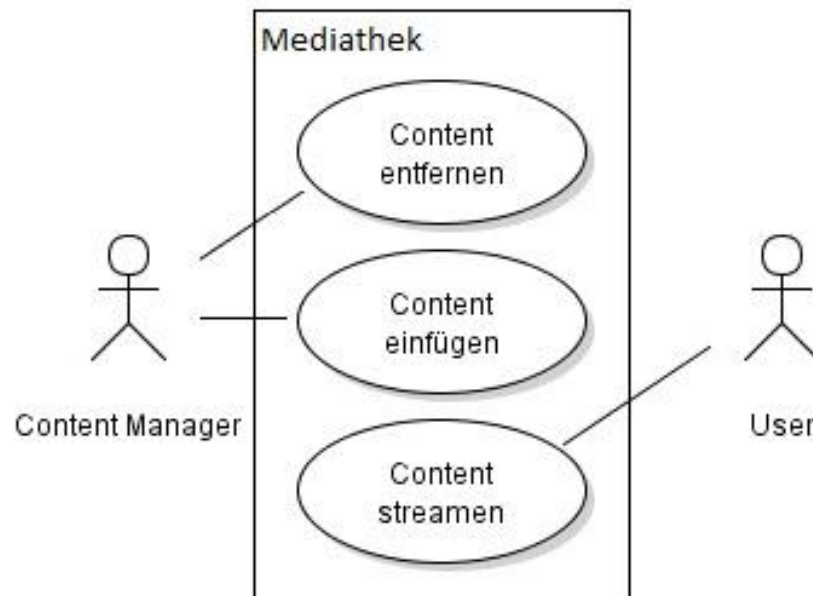
- **Aktoren bestimmen**
- **Anwendungsfälle bestimmen**
- **Anwendungsfall-Diagramme erstellen**
- **Anwendungsfälle beschreiben**



# Aufgabe 1 – Objektorientierte Analyse

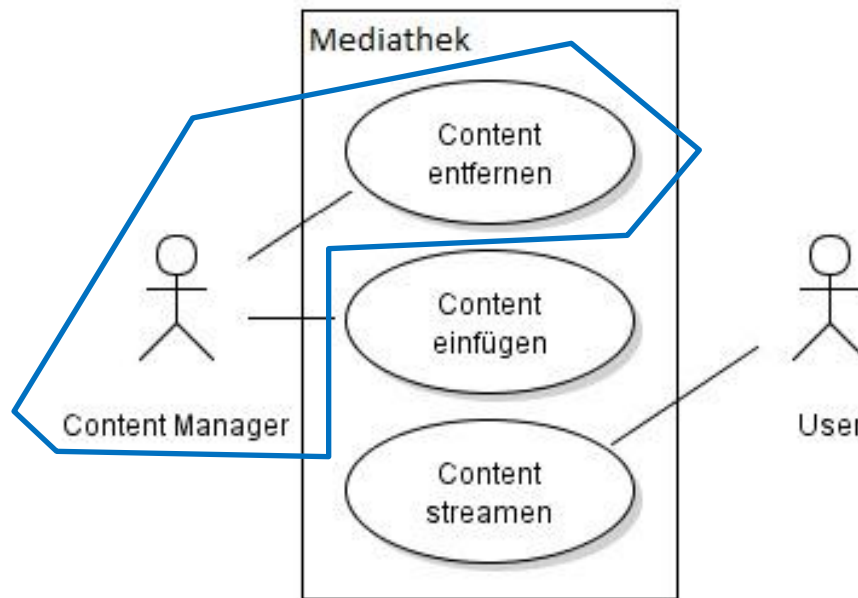
**Rollen: Content Manager, User**

**Use Case-Diagramm (Aktoren & Use Cases):**



# Aufgabe 1 – Objektorientierte Analyse

**Rollen: Content Manager, User**  
**Use Case-Diagramm (Aktoren & Use Cases):**



**Anwendungsfall beschreibt:**

- Anforderungen an System
- Interaktion zwischen Aktoren und System

**Anwendungsfall-Beschreibung enthält:**

- 1) Name
- 2) Kurzbeschreibung
- 3) Vorbedingung
- 4) Nachbedingung
- 5) Primärszenario
- 6) Mögliche Sekundärszenarien
- 7) Optional: Aktivitätsdiagramm

# Aufgabe 1 – Objektorientierte Analyse

**Name:** Content entfernen

**Kurzbeschreibung:** Ein Content Manager (CM) entfernt Content aus der Mediathek, so dass alle externen Nutzer keinen Zugriff mehr auf den Content haben.

**Vorbedingung:** Die Mediathek enthält Content der entfernt werden kann.

**Nachbedingung:** Der entfernte Content ist für externe Nutzer nicht mehr in der Mediathek sichtbar/ abrufbar.

**Primärszenario:**

1. CM gibt ContentID in Content Management System (CMS) ein
2. Content ist vorhanden und kann markiert werden
3. CM entfernt Content, so dass er in der Mediathek nicht mehr abrufbar ist

**Sekundärszenario:**

3a CM soll weiteren Content entfernen

3a1 weiter mit Schritt 1

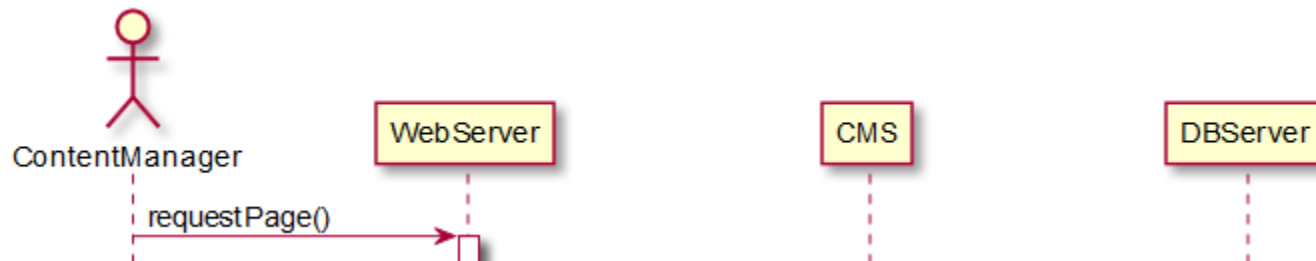
## Erstellung eines statischen Teilmodells:

- 1) Klassen identifizieren
  - 2) Assoziationen bestimmen
  - 3) Attribute identifizieren
  - 4) Vererbung einführen
  - 5) Modell überarbeiten
- iterativer Prozess
- Klassendiagramme

# Aufgabe 1 – Objektorientierte Analyse

Entwurf von Interaktionsdiagrammen:

- Identifiziere Nachrichten, die verschickt werden und Objekte, die Nachrichten verschicken
- Konstruiere Diagramme für jeden Anwendungsfälle

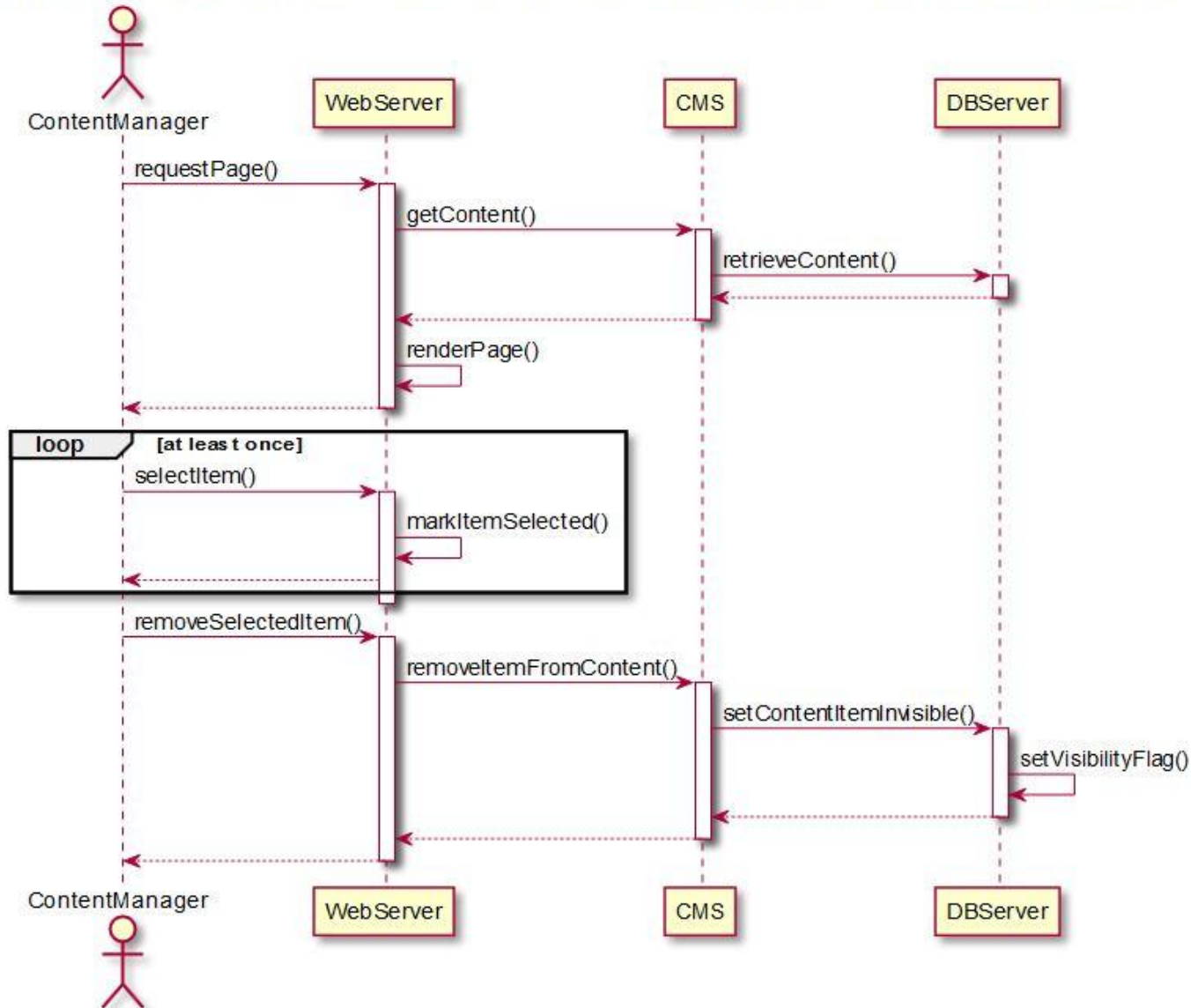


Objekt: WebServer, CMS, DBServer

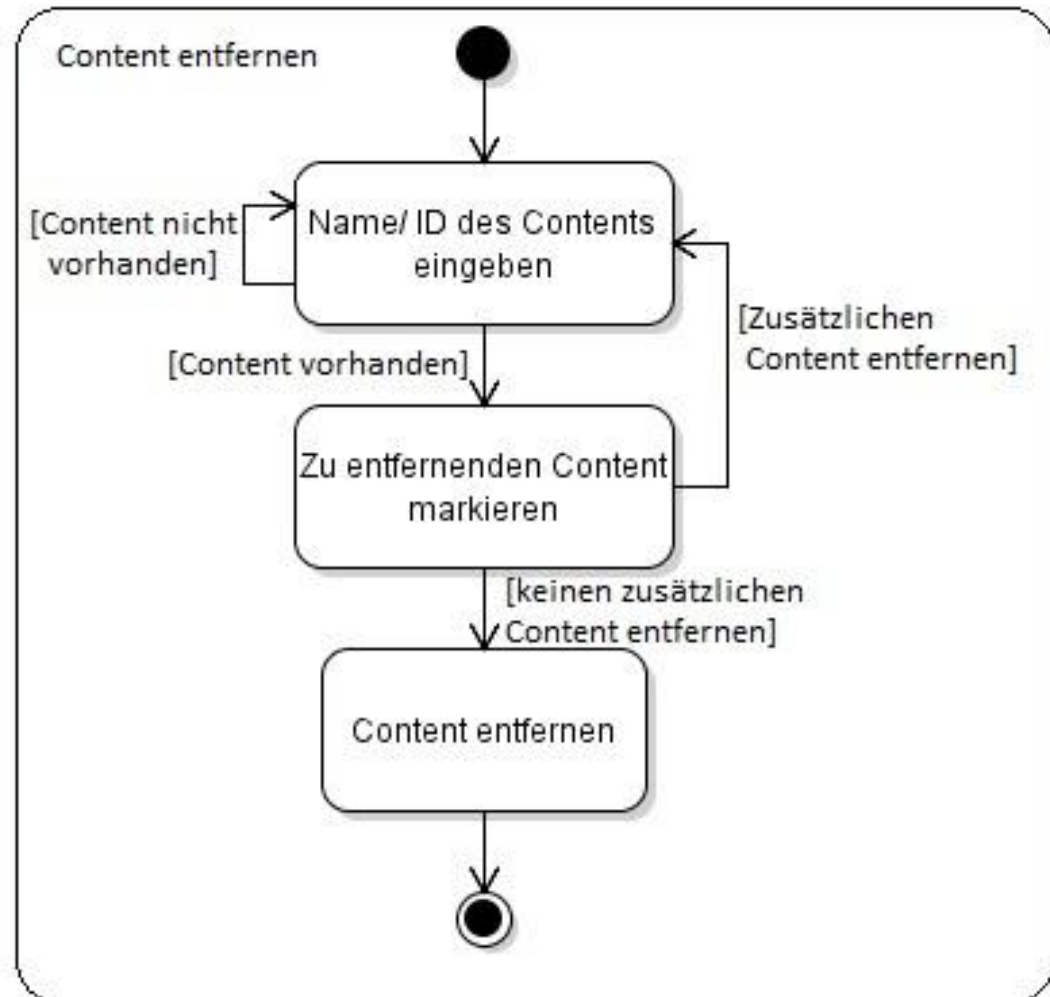
Nachrichten z.B.: requestPage(), renderPage(), removeSelectedItem(), ...

# Aufgabe 1 – Objektorientierte Analyse

Possible example demonstrating the interaction for the Use Case "Remove item from Content"



## Aktivitätsdiagramm: Erstellen aus SDs







**Am Ende der OOA steht...**

- **...ein lauffähiger Prototyp.**
- **...ein Systemmodell, das durch Diagramme beschrieben wird.**
- **...ein dynamisches Teilmodell, das Klassendiagramme enthält.**
- **...ein dynamisches und ein statisches Teilmodell, auf deren Basis ein Systementwurf erstellt wird.**

**Ausgangspunkt:**

**Ergebnisse der OOA (statisches/ dynamisches Teilmodell)**

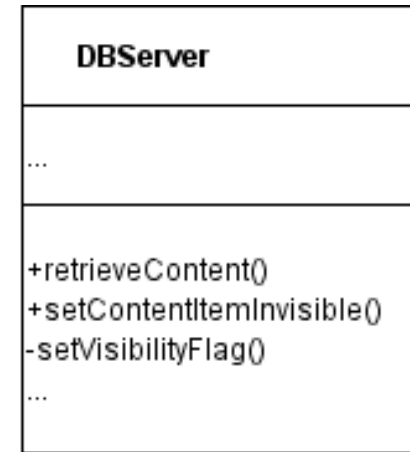
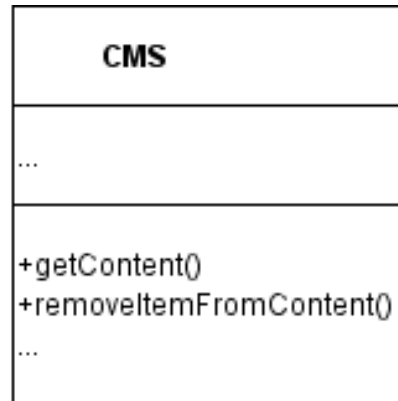
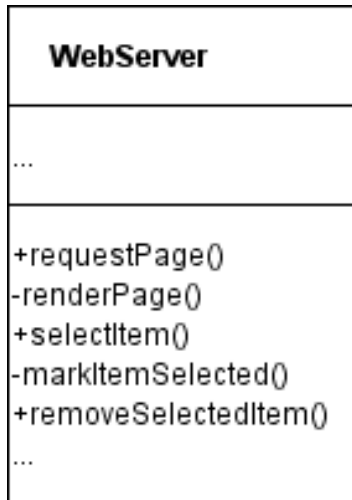
**Ziel:**

**Modell der Systemimplementierung durch**

- **Verfeinerung des Analysemodells**
- **Einbindung in die Systemumgebung durch Entwurf von Schnittstellen**
- **Konstruktion der Systemarchitektur**

# Aufgabe 2 – Objektorientiertes Design

Den Klassen werden die im Sequenzdiagramm verwendeten Methoden hinzugefügt, d.h. dass sie direkt in den Code eingefügt werden und den Klassen im Klassendiagramm hinzugefügt werden:



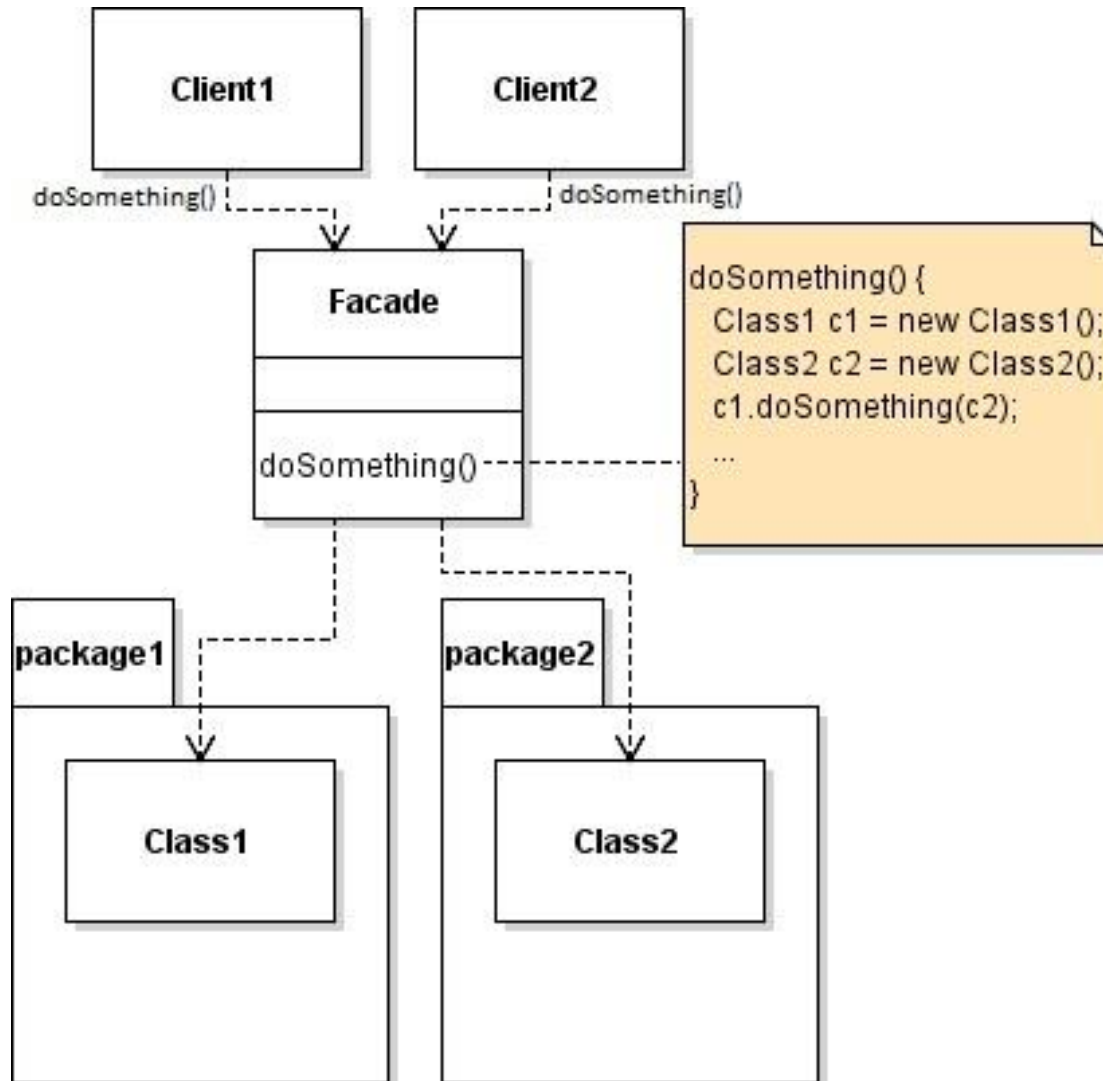
**Möglichst geringe Kopplung durch:**

**Facade Design Pattern (Fassaden Pattern)**

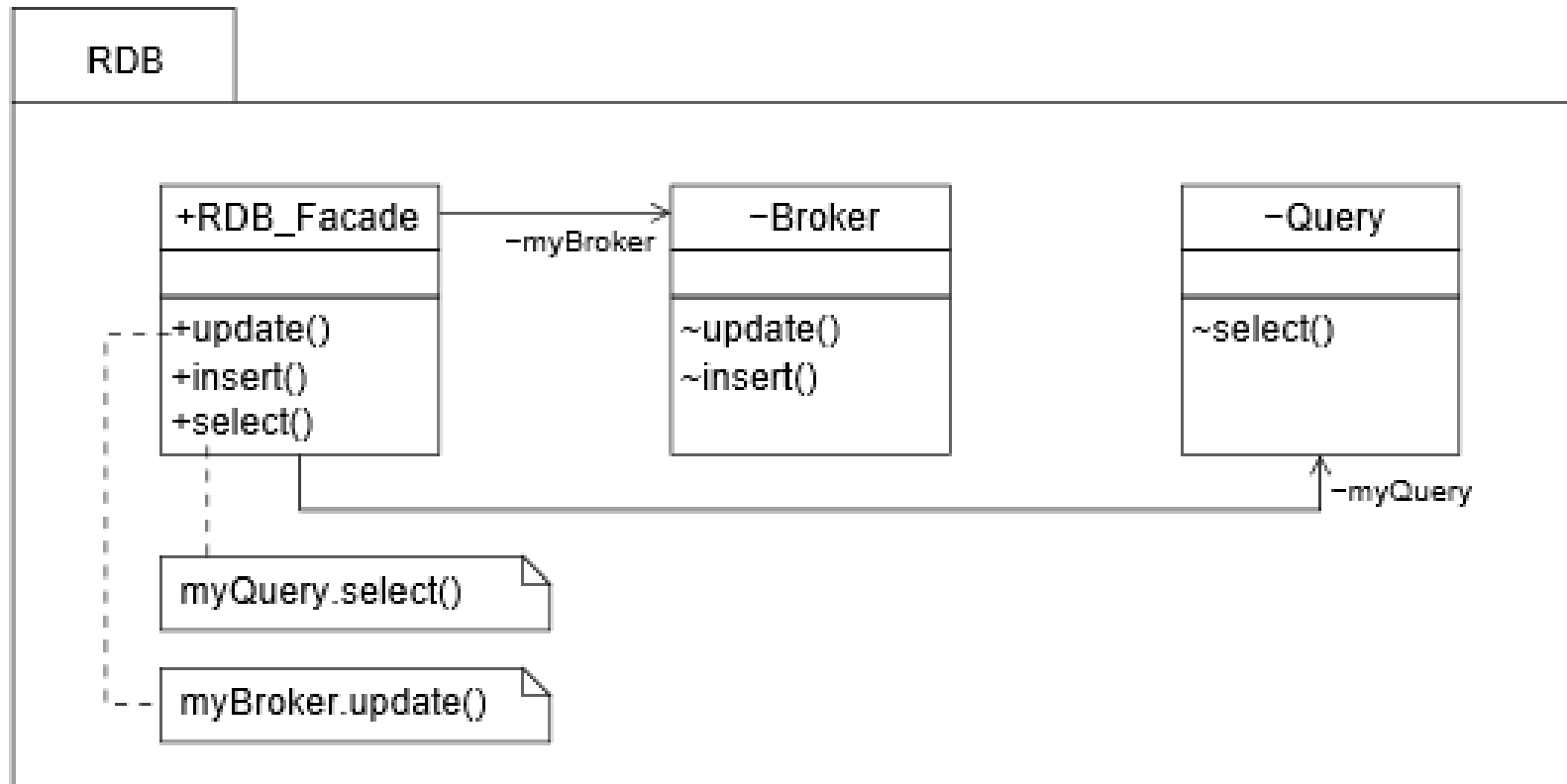
**→ Zusammenfassen verschiedener Dienste in einer Fassadenklasse**

- **Vorteil: lose Kopplung, geringere Komplexität nach außen**
- **Nachteil: zusätzliche Indirektionsstufe**

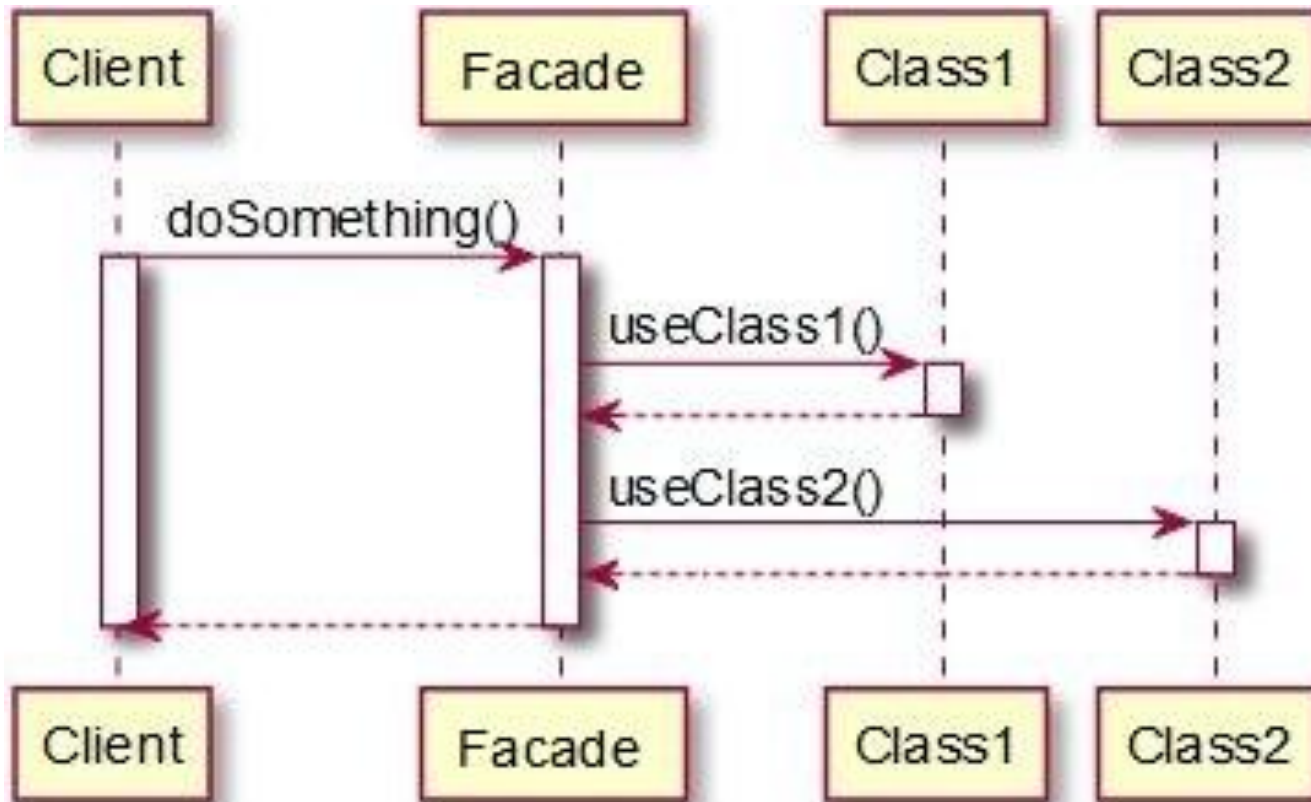
# Aufgabe 2 – Objektorientiertes Design



Aus der Vorlesung mit Fassadenklasse im gleichen Package:



# Aufgabe 2 – Objektorientiertes Design





**Am Ende des OOD steht...**

- **...ein lauffähiger Prototyp.**
- **...die SW-Architektur als Vorlage für die Implementierung.**
- **...das fertige Produkt.**
- **...ein dynamisches Teilmodell, das Sequenzdiagramme enthält.**