

# Vortragsthema

Johannes Pfann

Lehrstuhl für Software Engineering  
Friedrich-Alexander-Universität Erlangen-Nürnberg

# Gliederung

- 1 Verhaltensmuster
- 2 Observer
- 3 Command
- 4 Visitor
- 5 Zusammenfassung
- 6 Quellen

# Gliederung

- 1 Verhaltensmuster
  - Was sind Erzeugungsmuster
  - Typen von Erzeugungsmustern

2 Observer

3 Command

4 Visitor

5 Zusammenfassung

6 Quellen

# Verhaltensmuster – Was ist das?

## Verhaltensmuster ...

- Zuweisung von Zuständigkeiten
- Wechselseitige Kommunikation zwischen Objekten
- Beschreibung von komplexen Programmabläufen

## Konsequenzen

- Trennung von Verantwortlichkeiten
- Verringerung der Kopplung
- Erhöhung der Flexibilität der Software hinsichtlich ihres Verhaltens
- Bessere Verständlichkeit eines Programmablaufes

# Typen von Erzeugungsmustern

## Klassenbasiert

Klassenbasierte Verhaltensmuster wenden für die Verhaltenszuordnung zu den Klassen das Vererbungsprinzip an.

- Template Method
- Interpreter

# Typen von Erzeugungsmustern

## Objektbasiert

Bei objektbasierten Mustern wird der Erzeugungsprozess an andere Objekte delegiert

- Observer
- Command
- Visitor
- Strategy
- Mediator
- Iterator
- Memento
- State
- Chain of Responsibility

# Gliederung

## 1 Verhaltensmuster

## 2 Observer

- Definition
- Klassendiagramm - Observer Pattern
- Beispiel - Arbeitsvermittlung
- Implementierungsmöglichkeiten

## 3 Command

## 4 Visitor

## 5 Zusammenfassung

## 6 Quellen

# Definition

## Zweck

Definition einer 1-zu-n-Abhängigkeit zwischen Objekten, damit im Fall einer Zustandsänderung eines Objekts alle davon abhängigen Objekte entsprechend benachrichtigt und automatisch aktualisiert werden.

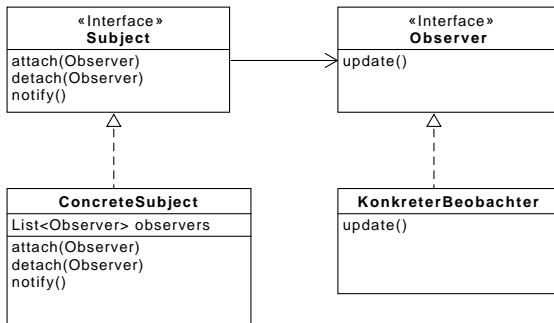
## Motivation/Ziel

- asdf
- asdf



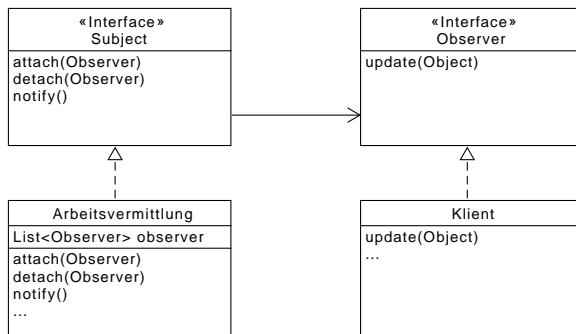
# Observer Pattern

- Basiert auf Rollen
- Stellt Mechanismus zur Broadcast-Kommunikation dar



# Arbeitsvermittlung

- 1-zu-n-Kommunikation zwischen Vermittlung und Klienten (Broadcast)
- Arbeitsvermittlung kennt weder Anzahl noch konkrete Klienten
- Klienten melden sich nur an, wenn sie daran interessiert sind.



# Beispiel - Klient

```
1 public interface Subject{
2     void attach(Observer aObserver);
3     void detach(Observer aObserver);
4     void notifyObservers();
5 }

1 public class Klient implements Observer {
2
3     public void update(Angebot aAngebot) {
4         if(aAngebot instanceof InformatikJob){
5             doSomething(aAngebot);
6         }
7     }
8 }
```

# Beispiel - Arbeitsvermittlung

```
1 public interface Observer{
2     void update(Object aObject);
3 }

1 public class Arbeitsvermittlung implements Subject {
2
3     List<Observer> mObserver = new LinkedList<Observer>();
4
5     public void attach(Observer aObserver) {
6         mObserver.add(aObserver);
7     }
8     public void detach(Observer aObserver) {
9         mObserver.remove(aObserver);
10    }
11    public void notifyObservers(Angebot aAngebot) {
12        for(Observer observer : mObserver){
13            observer.update(aAngebot);
14        }
15    }
16 }
```

# Implementierungsmöglichkeiten

## Push Modell

- Daten nur in der update-Methode
- Zugriff auf Subject nicht erlaubt
- Subjekt muss Interesse der Observer kennen.

## Pull Modell

- update-Methode ohne Parameter
- Zugriff auf Subjekt erwünscht
- Observer müssen Subject kennen

Beides kann auch gemischt werden!

# Implementierungsmöglichkeiten

## Observer beobachten mehrere Subjects

- Observer registriert sich bei mehreren Subjects
- Muss allerdings unterschiedlich darauf reagieren
- Lösung: erweiterung der update-Methode mit Subject

```
1 public void update(Subject aSubject, Angebot aAngebot) {  
2     if(aSubject instanceof ArbeitsvermittlungA) {  
3         doSomething(aAngebot);  
4     }  
5     if(aSubject instanceof ArbeitsvermittlungB) {  
6         doSomething(aAngebot);  
7     }  
8     ...  
9 }
```

# Implementierungsmöglichkeiten

## Ausführung der Updates durch Subject

- Weniger fehleranfällig
- Jedoch zu häufige Updates

## Ausführung der Updates durch Client

- Fehleranfälliger
- Regulierung der Updates

# Gliederung

1 Verhaltensmuster

2 Observer

**3 Command**

4 Visitor

5 Zusammenfassung

6 Quellen



# Gliederung

- 1 Verhaltensmuster
- 2 Observer
- 3 Command
- 4 Visitor**
- 5 Zusammenfassung
- 6 Quellen

# Gliederung

- 1 Verhaltensmuster
- 2 Observer
- 3 Command
- 4 Visitor
- 5 Zusammenfassung**
- 6 Quellen

# Gliederung

- 1 Verhaltensmuster
- 2 Observer
- 3 Command
- 4 Visitor
- 5 Zusammenfassung
- 6 Quellen**