



Softwaretechnik 2015/2016

PST Lehrstuhl

Prof. Dr. Matthias Hölzl

Joschka Rinke

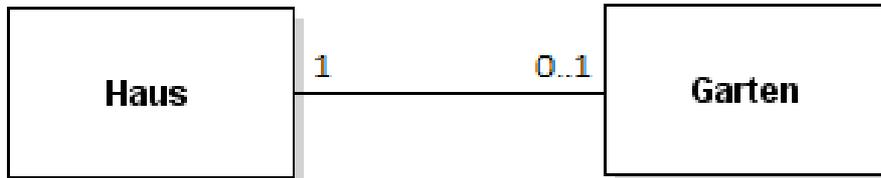


- Übung 6:**
- 19.11.2015**
- **Fragen**
 - **Klausurtermin: 10.02.2016 12-14 Uhr**
 - **Besprechung Blatt05**

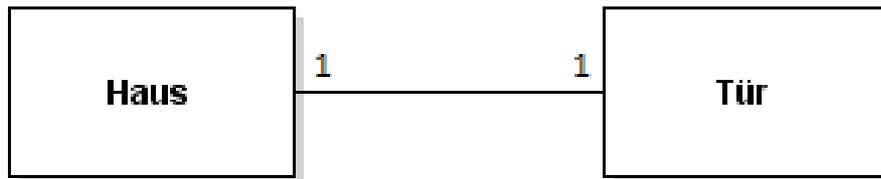


Klassendiagramm → Java-Implementierung:

- **Klassen: public class...**
- **Interface: public interface...**
- **Attribute: übernehmen und Sichtbarkeiten beachten**
- **Operationen: als Methoden zu Klasse/Interface hinzufügen**
- **Vererbung: Deklaration mit „extends“ und
Überschreibungen beachten (Konstruktor, abstrakte Methoden)**
- **Interface Implementierung: Deklaration mit „implements“ und
Methoden implementieren**
- **Assoziationen: Multiplizitäten beachten**



```
class Haus {private Garten garten;}
```



```
class Haus {private Tuer tuer = new Tuer();}
```



```
class Haus {private Set<Fenster> fenster =
new HashSet<Fenster>();}
```

Aufgabe 2 – Template Operation

Modellierungsmöglichkeiten um ähnliches Verhalten/ähnliche Klassen leicht im System hinzufügen zu können:

- **Interfaces (nur gemeinsame Methodensignaturen)**
- **Abstrakte Klassen (mindestens eine abstrakte Methode)**
 - **Interfaces als „Spezialfall“ von abstrakten Klassen**
- **Vererbung (Unterklasse „erbt“ alle Methoden/Attribute der Oberklasse)**

Aufgabe 2 – Template Operation



- Das System wird so umgebaut, dass die Klasse Vorstellung...**
- ...als Interface implementiert wird.**
- ...als abstrakte Klasse implementiert wird.**
- ...nicht verändert wird und alle anderen Vorstellungsarten von ihr erben.**
- ...entfernt wird und durch entsprechende unterschiedliche Vorstellungsarten ersetzt wird.**



Vorteile der Verwendung von abstrakten Klassen sind?

- **Weniger Klassen im System und damit mehr Übersichtlichkeit**
- **Vermeidung von redundantem Code**
- **Mehr Struktur indem ähnliche Klassen als Unterklassen einer abstrakten Klasse implementiert werden**
- **Vermeidung der Verwendung von Interfaces**

Aufgabe 2 – Template Operation

Vorteile durch Verwendung von „Template Method Pattern“?

- **Unterklassen können ähnliche Methoden entsprechend unterschiedlich implementieren**
- **Vermeidung von doppeltem und somit redundantem Code**
- **Kontrolle darüber, was verändert wird, nämlich am besten nur genau die abstrakten Methoden der (abstrakten)Oberklasse**



Hat ein Attribut einer Klasse die Sichtbarkeit protected, so ist es...

- **...nur in allen Subklassen der Klasse sichtbar.**
- **...überall sichtbar.**
- **...nur im gleichen Package sichtbar.**
- **...nur im gleichen Package und in allen Subklassen sichtbar**
- **...nur innerhalb der Klasse sichtbar**

Aufgabe 2 – Template Operation

Welche Sichtbarkeiten gibt es?

- **public (+):** überall sichtbar
- **private (-):** nur in der Klasse sichtbar
- **protected (#):** nur im gleichen package und in allen Subklassen sichtbar
- **default (~):** nur im gleichen package sichtbar

Welche Attribute und Operationen sind wo sichtbar?

public: reservieren(...), oeffnen(), schliessen()

private: name, anzahlPlaetze, nummer, reihe, alter

protected: openAir

Default (package): -

Aufgabe 2 – Template Operation

Attribute hinzufügen:

Welche Sichtbarkeit ist für Attribute in einer abstrakten Klasse (meist) sinnvoll?

→ protected, da dann die Attribute auch in den Subklassen sichtbar sind

Also:

- **protected String datum → Veranstaltung**
- **protected String uhrzeit → Veranstaltung**
- **private boolean beamerOn → Kinoveranstaltung**

Aufgabe 2 – Template Operation

Operationen hinzufügen:

In welcher Klasse sind welche Operationen sinnvoll?

- **starten: Veranstaltung**
- **beenden: Veranstaltung**
- **zeigeWerbung: Kinoveranstaltung**
- **zeigeFilm: Kinoveranstaltung**
- **setBeamer: Kinoveranstaltung**
- **getOpenAir: Veranstaltung**

Aufgabe 2 – Template Operation

Wie wird die Klasse Theatervorstellung am besten hinzugefügt?

Theatervorstellung realisiert ebenfalls Vorstellung!

Sinnvolle Attribute:

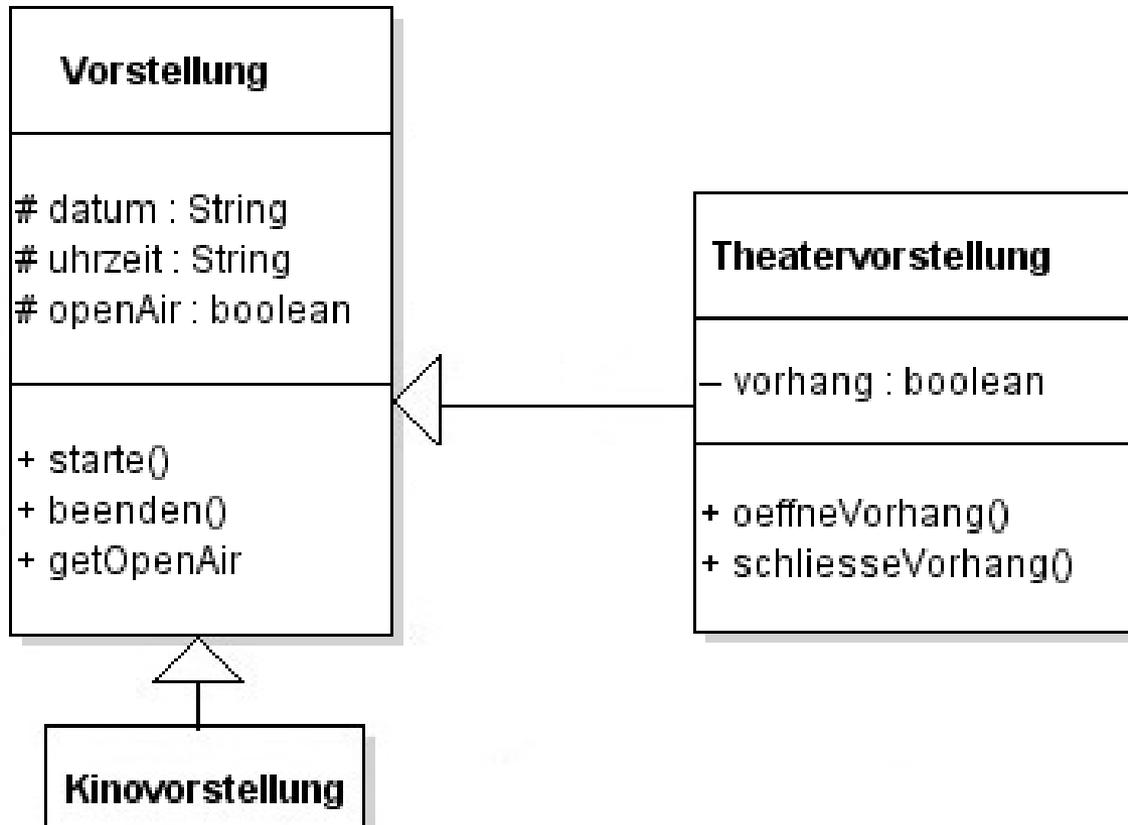
- vorhang : boolean

Sinnvolle Operationen:

+ vorhangOeffnen()

+ vorhangSchliessen()

Aufgabe 2 – Template Operation



Machen evtl. weitere Änderungen Sinn?

Aufgabe 2 – Template Operation

