



Softwaretechnik 2015/2016

PST Lehrstuhl

Prof. Dr. Matthias Hölzl

Joschka Rinke



Übung 7:
26.11.2015

- **Fragen**
- **Besprechung Blatt06**



Interaktionsdiagramme modellieren...

- ...die Interaktionen zwischen den Methoden einer Klasse
- ...die Interaktionen zwischen den Klassen eines Packages
- ...die Interaktionen zwischen mehreren Objekten
- ...das Verhalten eines Objekts



Diagramme zur Modellierung von dynamischen Verhalten:

- **Interaktionsdiagramme:**
 - Kommunikation & Zusammenarbeit mehrerer Objekte
 - Sequenzdiagramm, Kommunikationsdiagramm
- **Zustandsdiagramme:**
 - Verhalten eines Objekts einer Klasse zur Laufzeit
 - flach, hierarchisch, parallel
- **Aktivitätsdiagramme:**
 - (evtl. parallele) Abläufe von Aktivitäten



Bei MVC steht...

- ...das **M** für **Model** und repräsentiert die Daten und Änderungen auf den Daten
- ...das **M** für **Multiple** und besagt, dass mehrere Objekte miteinander interagieren
- ...das **C** für **Controller**, der Interaktionen des Anwenders mit der **View** in Aktionen übersetzt
- ...das **C** für **Control** und suggeriert, dass Anwender die Kontrolle über das Modell haben und die **View** ändern können



Model-View-Controller (MVC):

- **Model:** repräsentiert die Daten und Updates auf den Daten
- **View:** Rendert den Inhalt des Modells und muss upgedated werden, wenn sich das Model ändert
- **Controller:** Übersetzt die Interaktionen des Anwenders mit der View in Aktionen die das Model ausführt

Aufgabe 1 – MVC & Java Swing

MVC in der Vorlesung:

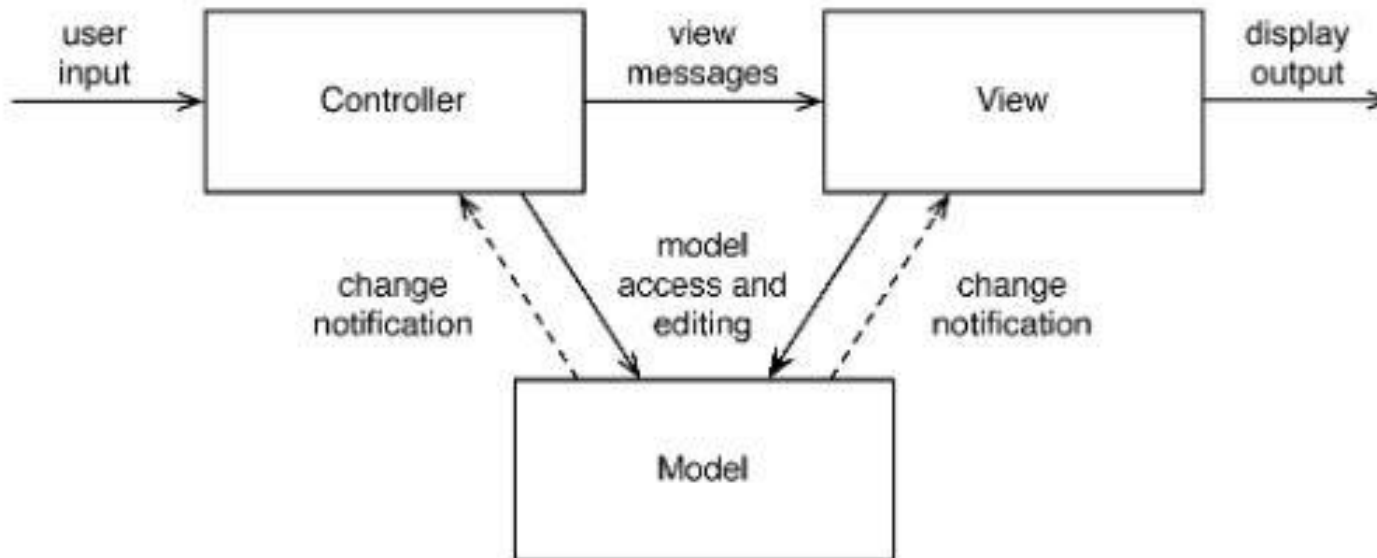
- klare Trennung zwischen View und Controller
- Controller und View als Observer,
die Change Notifications vom Modell erhalten,
d.h. sie müssten sich dort z.B. als Observer registrieren...

MVC in Java-Swing:

- keine klare Trennung zwischen View und Controller mehr,
es gibt eine Art ViewController (single User-Interface)

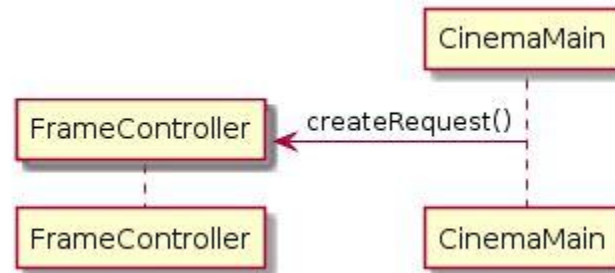
Java-Swing basiert also nur lose auf dem traditionellen MVC-Pattern
(In der VL wurde die Web-MVC Architektur als weitere Variante vorgestellt)

Aufgabe 1 – MVC & Java Swing



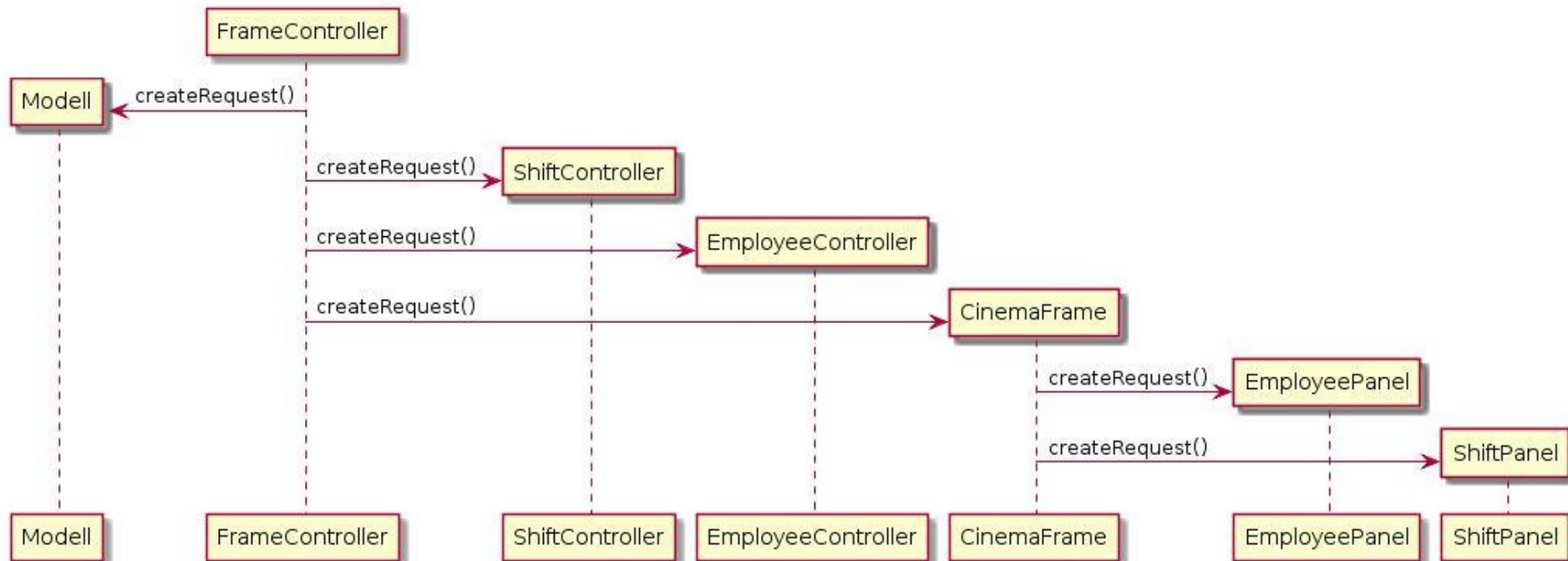
Aufgabe 1 – MVC & Java Swing

Möglicher Klassischer MVC Ansatz:



Vom FrameController-Objekt aus werden das Modell, die benötigten Controller und die Views instantiiert

Möglicher Klassischer MVC Ansatz:



Problem:

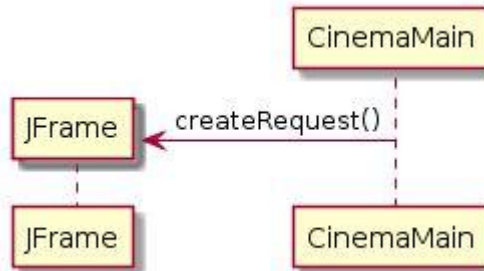
Alle Controller und alle Frames/Panels müssten Observer sein, die vom Modell über Änderungen upgedated werden.

Das ist in Swing nicht vorgesehen...

jetzt müsste noch das Observer-Interface implementiert werden ☹️

Aufgabe 1 – MVC & Java Swing

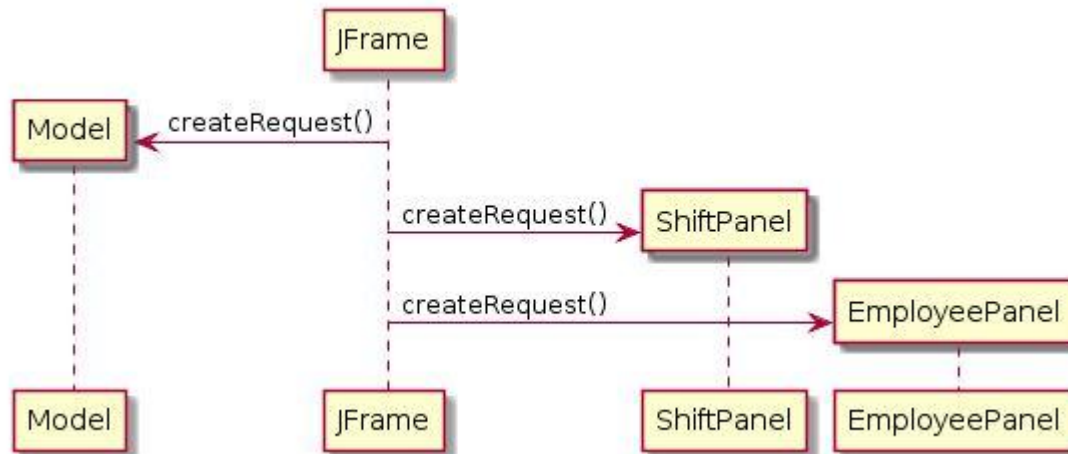
Möglicher Java Swing Ansatz:



Vom JFrame-Objekt aus wird das Modell und evtl. benötigte JPanel's instantiiert

Aufgabe 1 – MVC & Java Swing

Möglicher Java Swing Ansatz:



In den Panels werden für die Buttons z.B. ActionListener geaddet und die ActionEvents gehandled, das würde im klassischen MVC Ansatz in den Controllern geschehen.