

# Programmiermethodik

## Übung 7

Sommersemester 2011  
Fachgebiet Software Engineering

Andreas Scharf  
[andreas.scharf@cs.uni-kassel.de](mailto:andreas.scharf@cs.uni-kassel.de)

# Agenda

- **Organisatorisches**
- **Besprechung HA 4**
- **Model – View – Controller (MVC)**
- **Vorstellung HA5**
- **Praktische Übung: MVC üben**

# Hinweise zu Abgaben

- **Eclipse Projekte ohne .java Dateien können nicht bewertet werden!**
  - Testet euer exportiertes Projekt (am besten neuen Workspace nehmen)

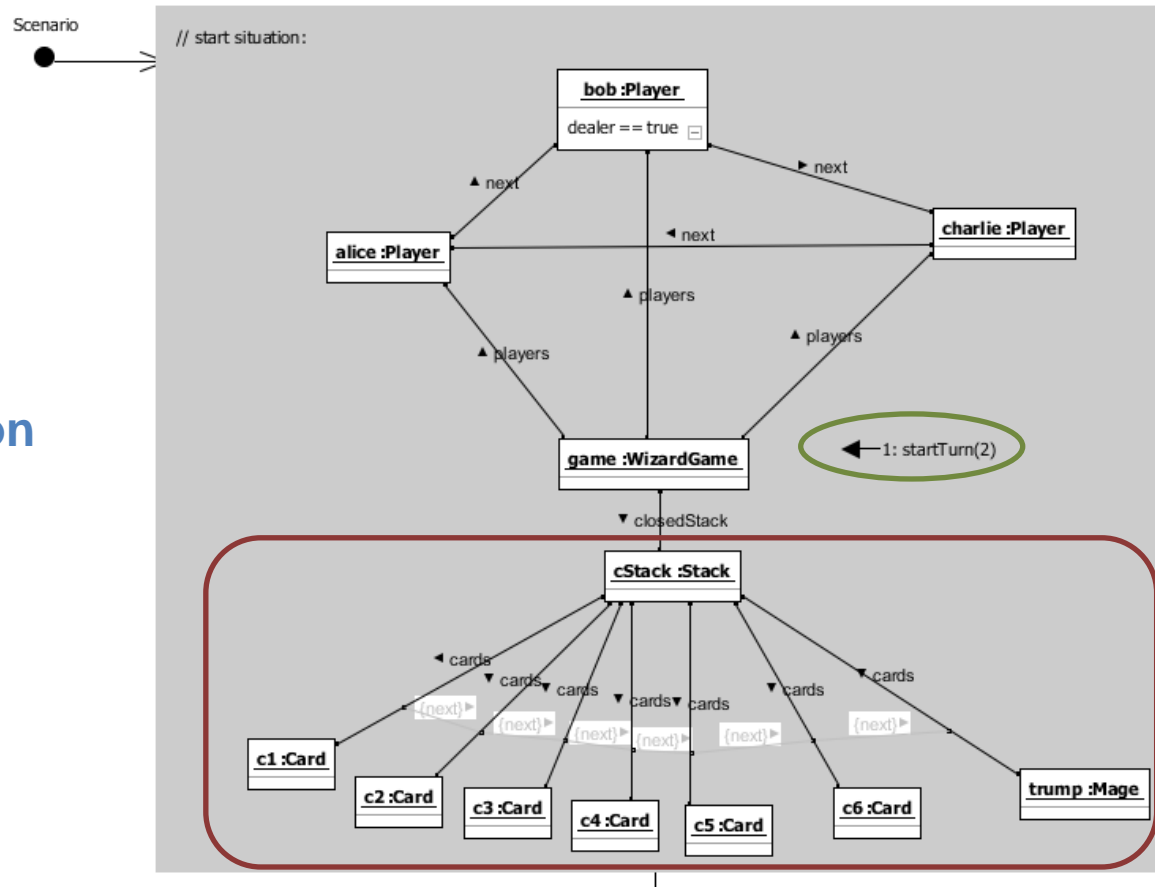
# Besprechung HA4 I

- **Storyboards mit Fujaba4Eclipse:**
  - Runde 2 initialisieren, Normaler Trumpf
  - Runde 2 initialisieren, Zauberer Trumpf
  - Karte ausspielen: Kann Farbe bedienen, bedient Farbe
  - Karte ausspielen: Kann Farbe bedienen, bedient nicht
  - Karte ausspielen: Kann Farbe nicht bedienen, wirft andere Farbe ab
  - Stich evaluieren: Höchste Karte
  - Stich evaluieren: Höchster Trumpf
  - Stich evaluieren: 2 Narren
  
- **Randbedingungen: 3 Spieler (Alice, Bob, Charlie)**

# Besprechung HA4 II

- Storyboard 1: Runde 2 initialisieren, Zauberer als Trumpf

Startsituation



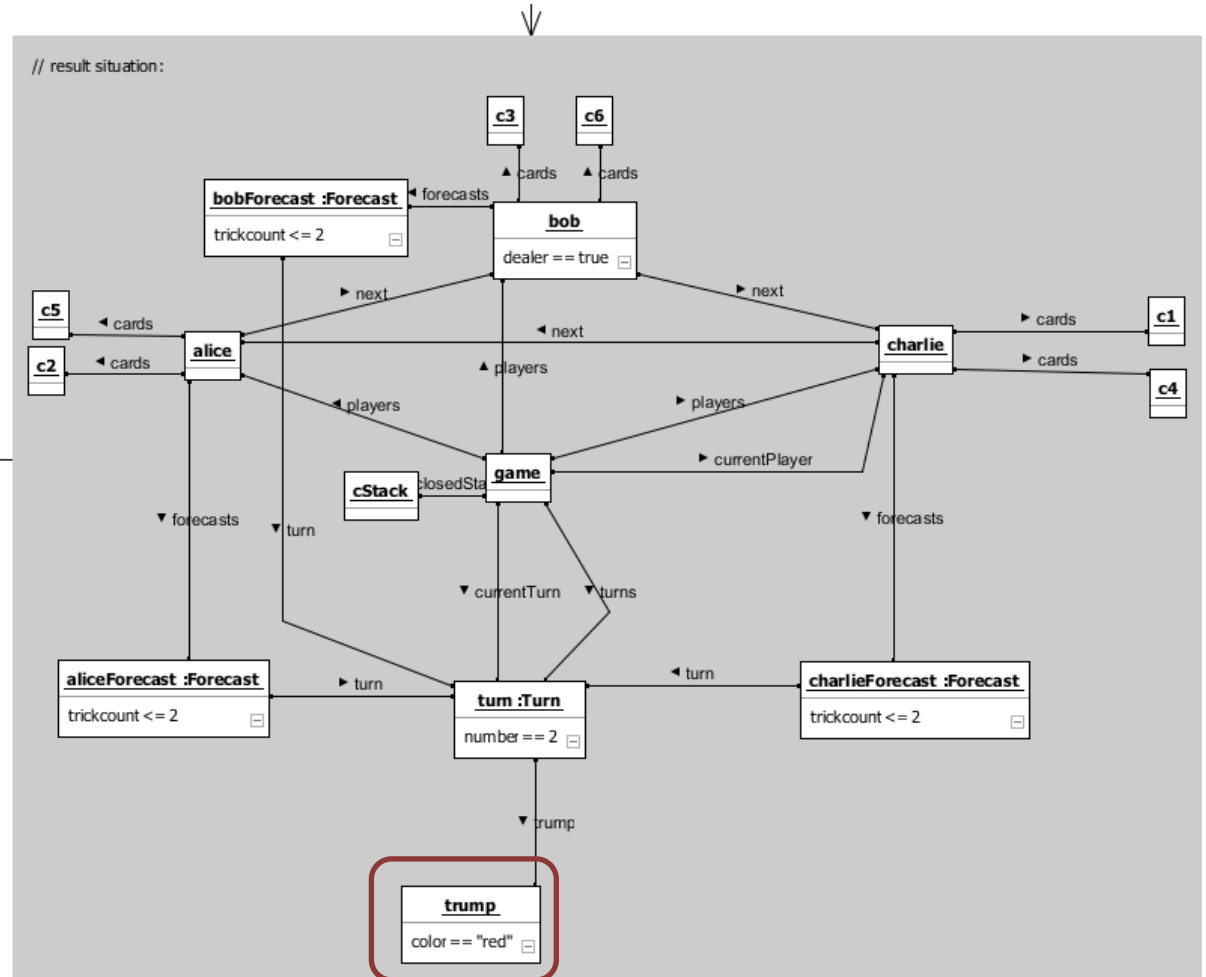
# Besprechung HA4 III

- Storyboard 2: Runde 2 initialisieren, Zauberer als Trumpf

Endsituation

Wichtig:

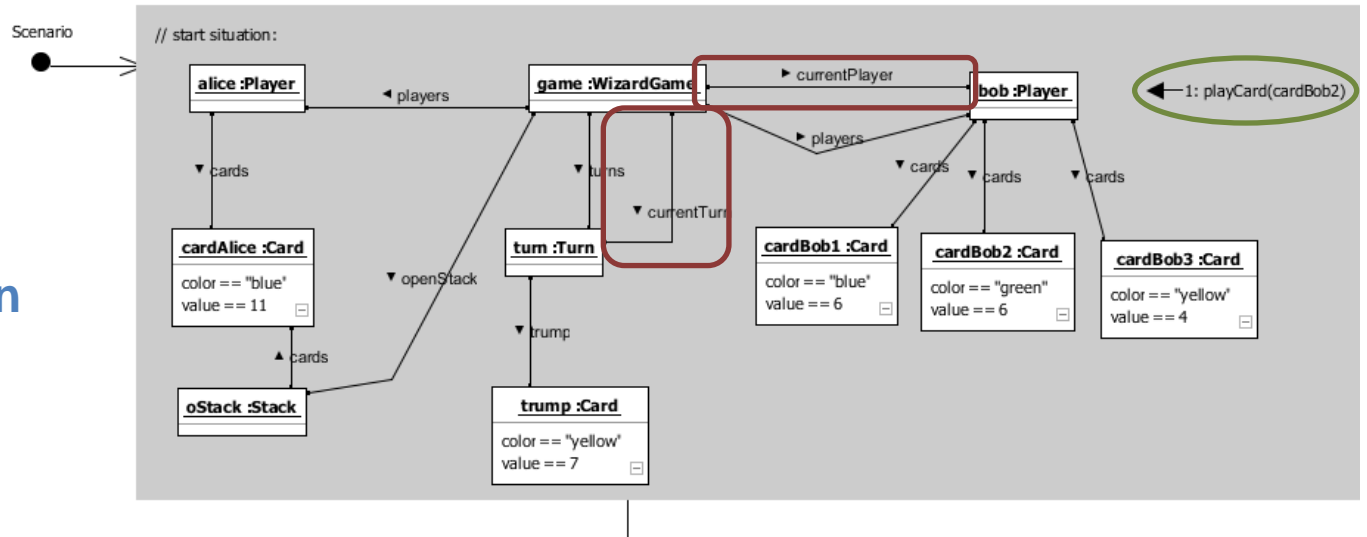
Card Objekte sind die gleichen wie in der Startsituation!



# Besprechung HA4 IV

- Storyboard 4: Kann Farbe bedienen, bedient nicht

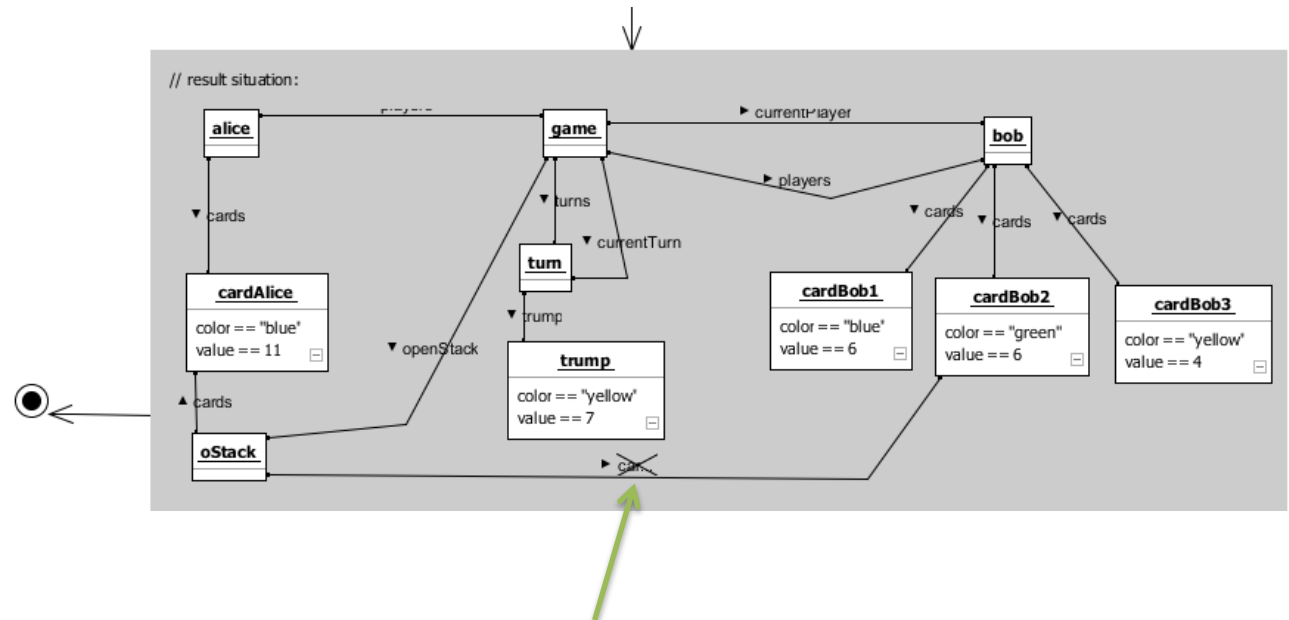
Startsituation



# Besprechung HA4 V

- Storyboard 4: Kann Farbe bedienen, bedient nicht

Endsituation



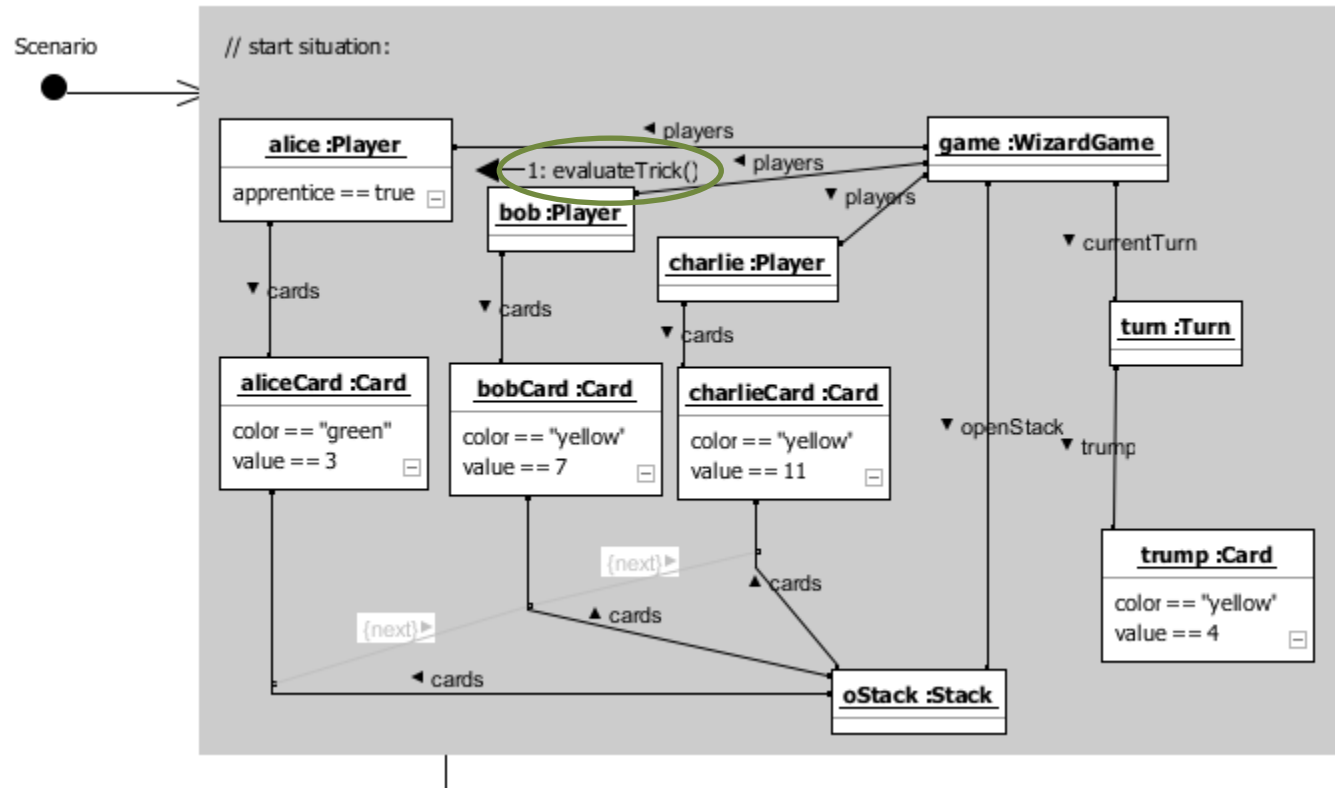
Info: KEIN Abzug falls der negative-link fehlt!



# Besprechung HA4 VI

- Storyboard 7: Höchster Trumpf gewinnt

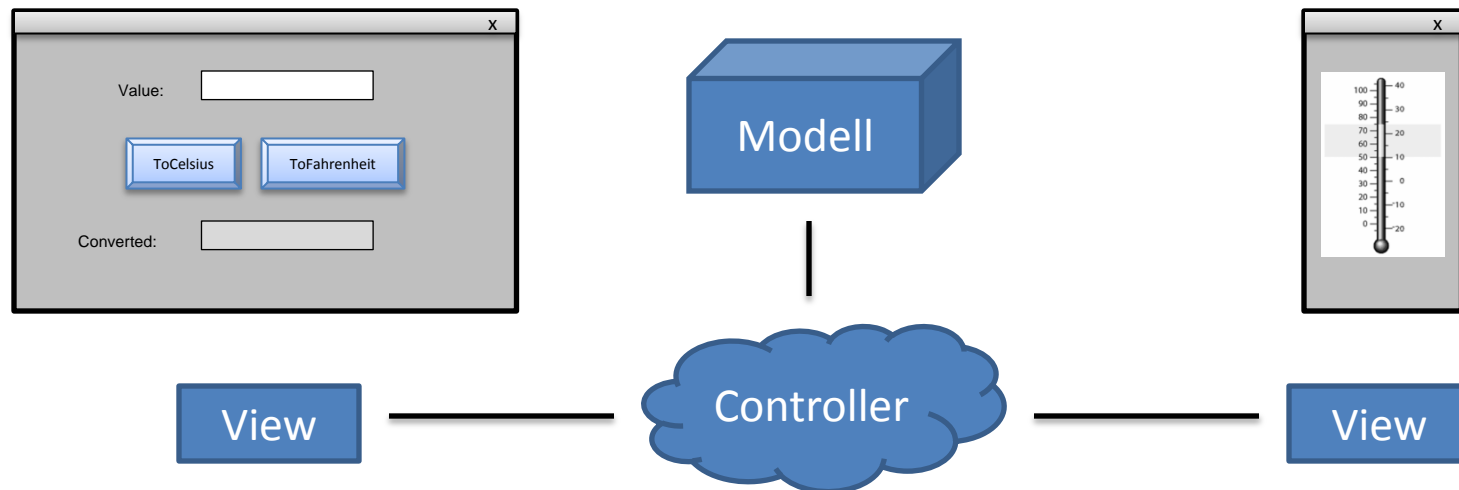
Startsituation





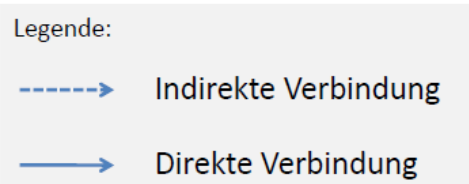
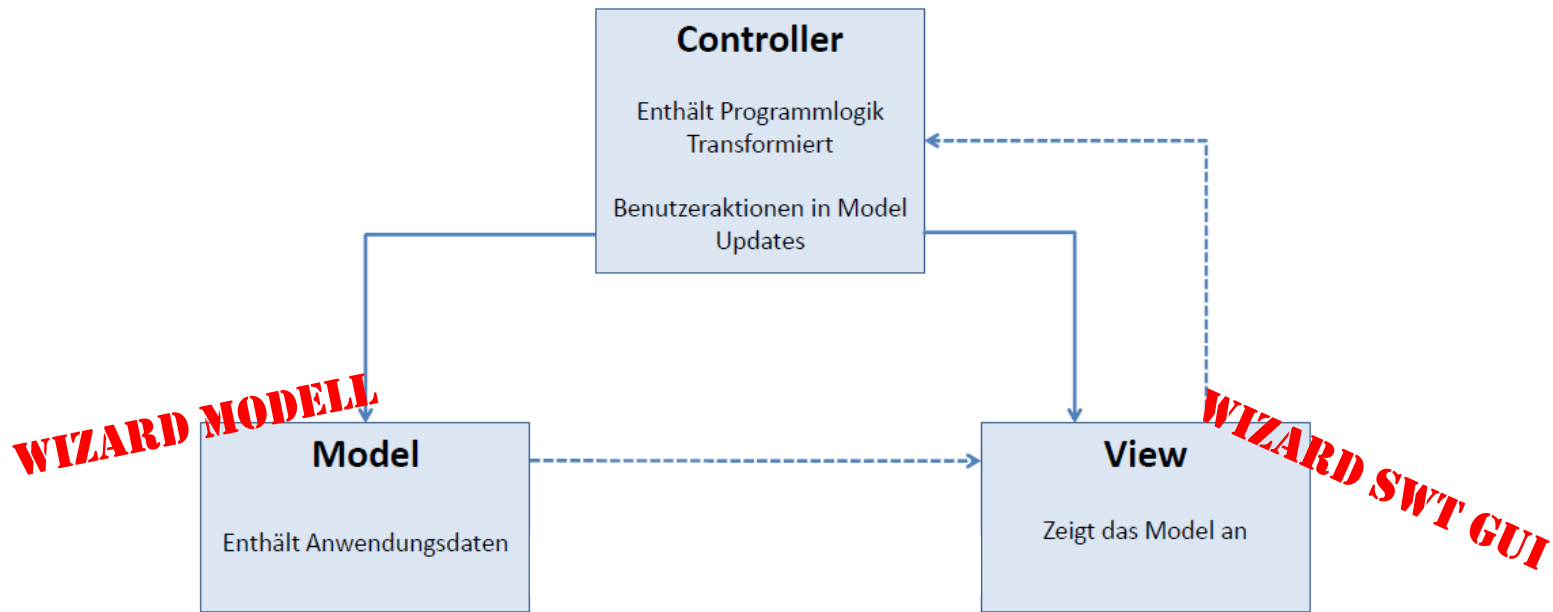
# Model – View – Controller (MVC) I

- **MVC ist ein Entwurfsmuster**
  - Entwurfsmuster lösen immer wiederkehrende Softwaretechnische Probleme
- **Entkoppelt die Anwendungslogik von ihrer Darstellung**
- **Vorteile**
  - Modell und Logik müssen nur einmal programmiert werden
  - Verschiedene (gleichzeitige) Darstellungen für dasselbe Modell



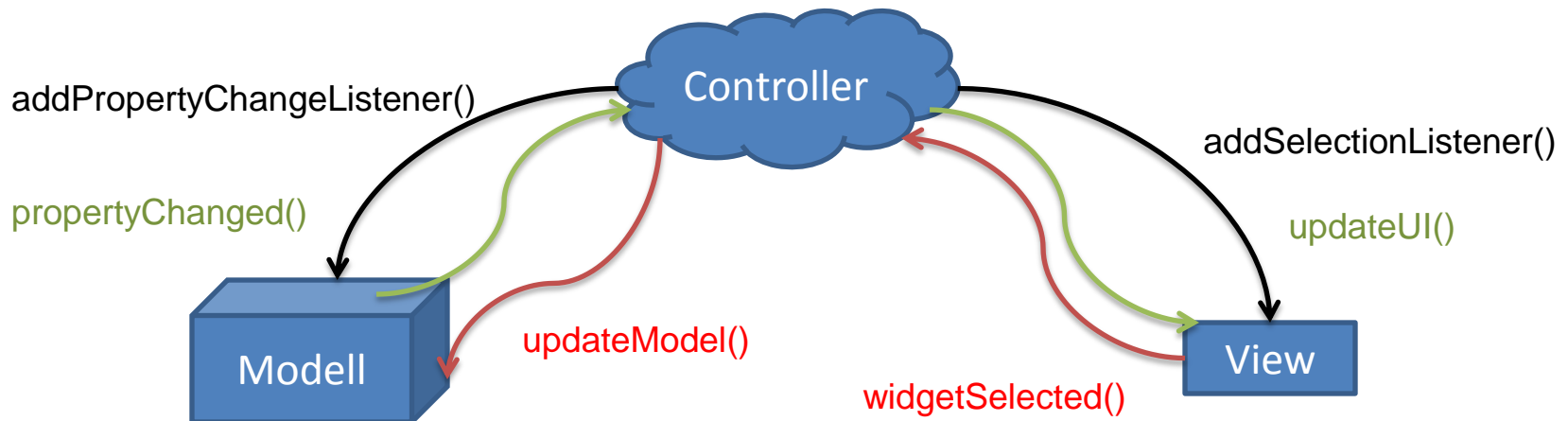
# MVC II

- MVC Überblick
- Genaue Implementierung variiert häufig



# MVC III

- Controller meldet sich als Listener an Modell und View an
- Controller empfängt User-Event (z.B. Button „XYZ“ gedrückt)
- Controller verändert Modell
- Modell informiert seine PropertyChangeListener (u.a. den Controller)
- Controller updated UI

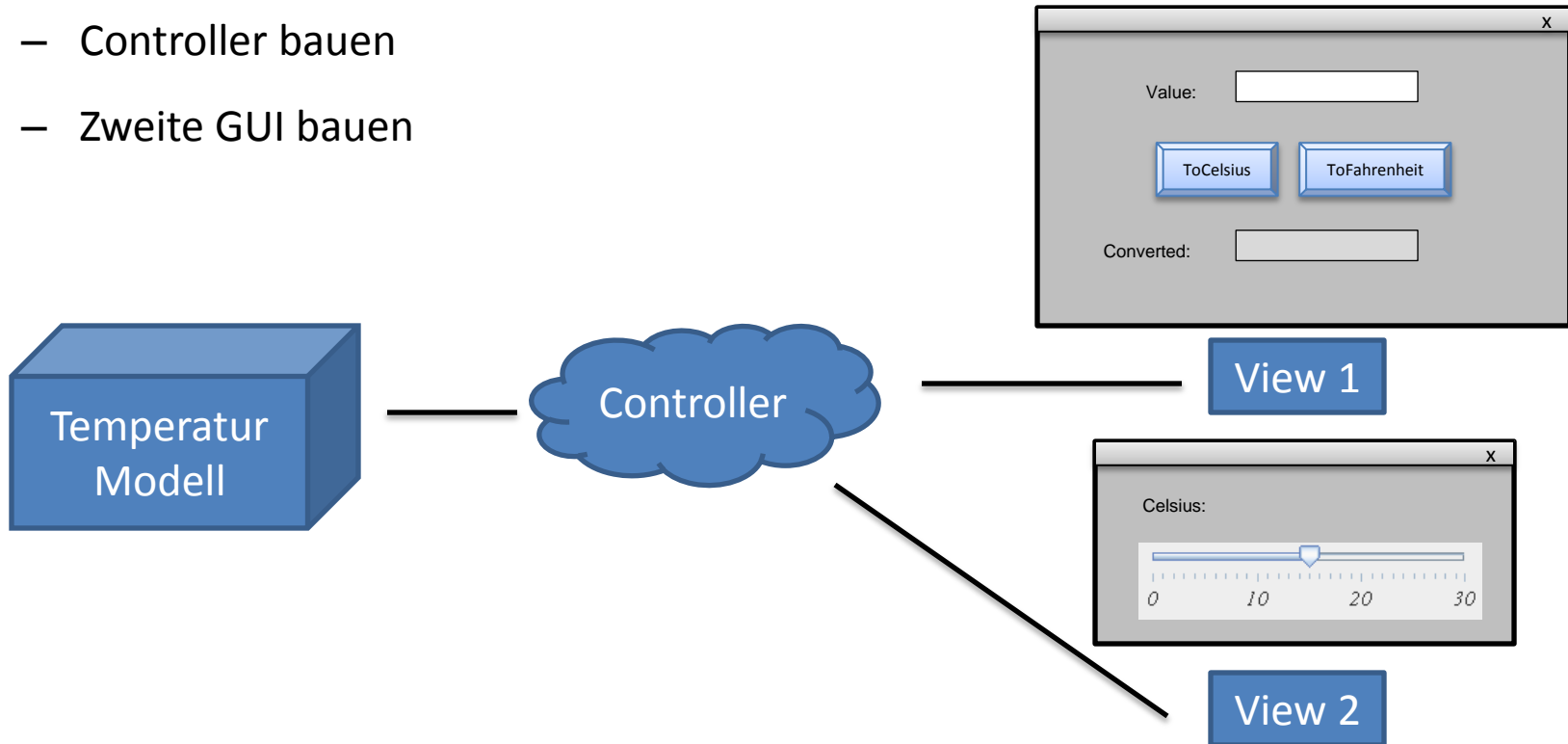


# MVC IV

- **Problem: Modell und View müssen Mechanismen bereitstellen um Listener anzumelden**
- **Bei den meisten GUI Bibliotheken kein Problem!**
- **Modell: Bei Settern immer angemeldete Listener informieren**
  - Viel Aufwand
  - Immer das gleiche
  - Warum nicht einfach mitgenerieren lassen? → Fujaba JavaBean Stereotyp
- **<<JavaBean>> Stereotyp**
  - An jede Klasse an der man PropertyChangeListener anmelden will
  - `addPropertyChangeListener(PropertyChangeListener listener)`
  - `removePropertyChangeListener(PropertyChangeListener listener)`

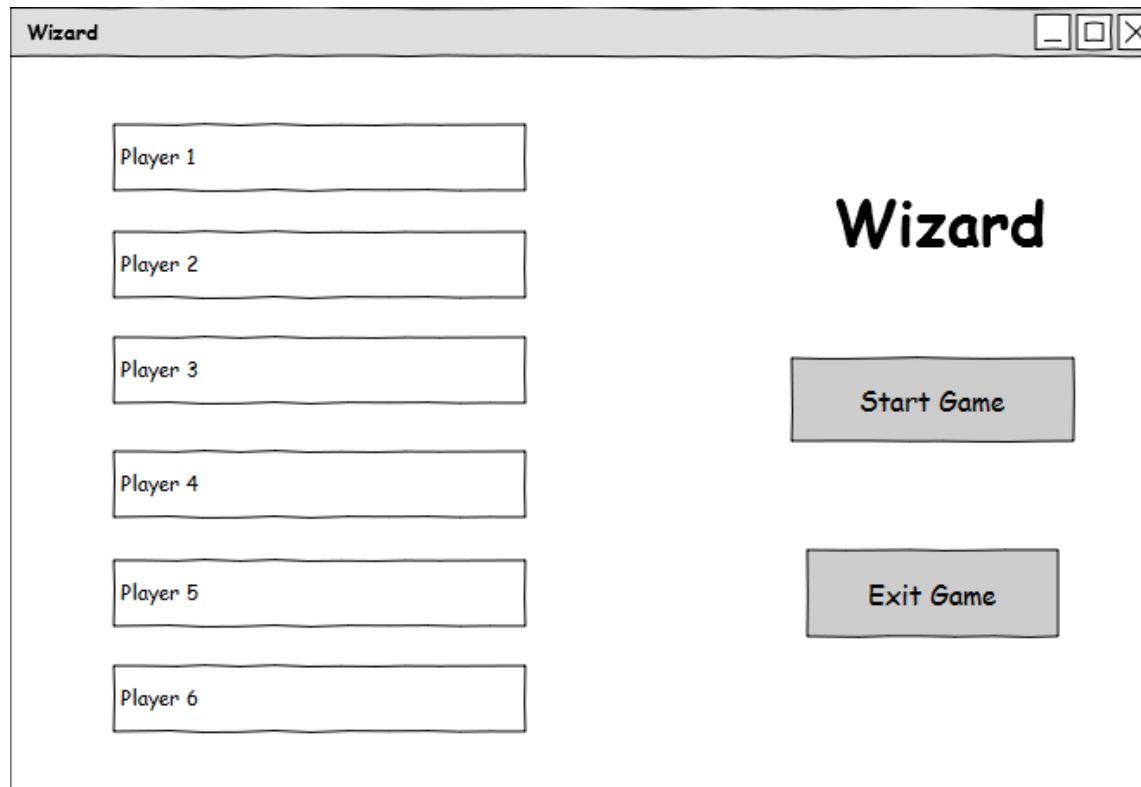
# MVC V

- **Mit CelsiusFahrenheitConverter üben**
  - „Richtiges“ Temperaturmodell erstellen
  - <<JavaBean>> Stereotyp hinzufügen
  - Controller bauen
  - Zweite GUI bauen



# Vorstellung HA 5 I

- Wichtig: KEIN „Absolute layout“ verwenden
- Aufgabe 1: Wizard Startscreen mit WindowBuilder modellieren






# Vorstellung HA 5 II


- Aufgabe 2: Wizard Gamescreen mit WindowBuilder modellieren

Wizard
\_ □ ×

Ablage








Trump  
Green




Current Tricks: 1

Forecast: 2

Alice' Turn



	Alice		Bob		Charlie	
Turn	Points	Forecast	Points	Forecast	Points	Forecast
1	20	0	-10	1	30	1
2	40	0	40	2	50	0
3	70	1	20	2	80	1
4	50	2	50	1	120	2
5	90	2	70	0	100	2
6		2		3		2
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

# Vorstellung HA 5 II

- Zusatzaufgabe: Karten überlagernd anordnen lassen

Wizard
☐ ☐ ☒

Ablage



Trump  
Green



Current Tricks: 1

Forecast: 2



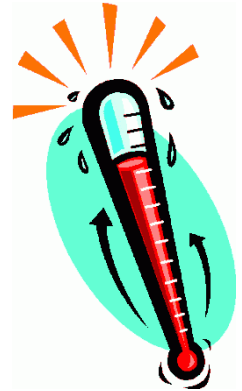
Alice' Turn

Turn	Alice		Bob		Charlie	
	Points	Forecast	Points	Forecast	Points	Forecast
1	20	0	-10	1	30	1
2	40	0	40	2	50	0
3	70	1	20	2	80	1
4	50	2	50	1	120	2
5	90	2	70	0	100	2
6		2		3		2
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

# Praktische Übung: MVC üben I

- **Vorbereitung**

- Converter Projekt vom Blog herunterladen
- Thermo.zip vom Blog herunterladen
- Projekt importieren
- Neues SWT/JFace Projekt anlegen
- Dateien vom Converter Projekt in das neue SWT/Jface Projekt kopieren
- Thermo.zip in nach lib/Thermo.zip kopieren
- Auf Thermo.zip: Rechtsklick -> Build Path -> Add to Build Path

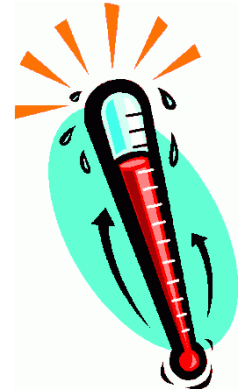


- **JavaBean Stereotyp an Temperature Klasse -> generieren**

- **Controller-Baustein schreiben**

# Praktische Übung: MVC üben II

- **JavaBean Stereotyp an Temperature Klasse -> generieren**
- **Controller-Baustein schreiben:**
  - Meldet sich als PropertyChangeListener am Modell an
  - Meldet sich als SelectionListener an der GUI an
  - Ändert das Modell falls der Benutzer auf „To...“ drückt
  - Refreshed die GUI bei Modelländerung
- **Slider GUI entwickeln und anbinden**



**Ende**

**Schönes WE!**