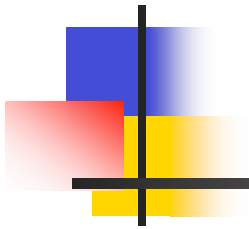
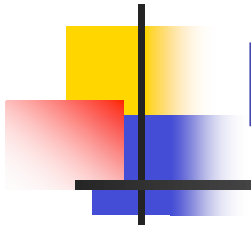


Software Engineering

Besprechung zur Uebung 4

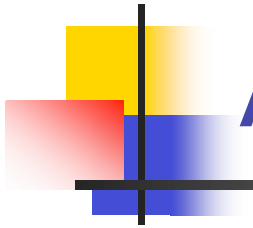
Architektur, Entwurf



Formelles, Allgemeines

- Gruppen
  - 4 Einzelgruppen derzeit.
- Form
  - Abgabe eines Dokuments (pdf)
  - Namen und Matr.Nummern auf das Deckblatt
  - Keine Umlaute oder Leerzeichen in den Dateinamen
- Lösungen
  - Allgemein waren die meisten Übungen gut gelöst
  - Teilweise Schwächen bei Architekturstilen / Entwurfsmustern
  - Zusatzaufgabe 2.6 wurde von keiner Gruppe gelöst

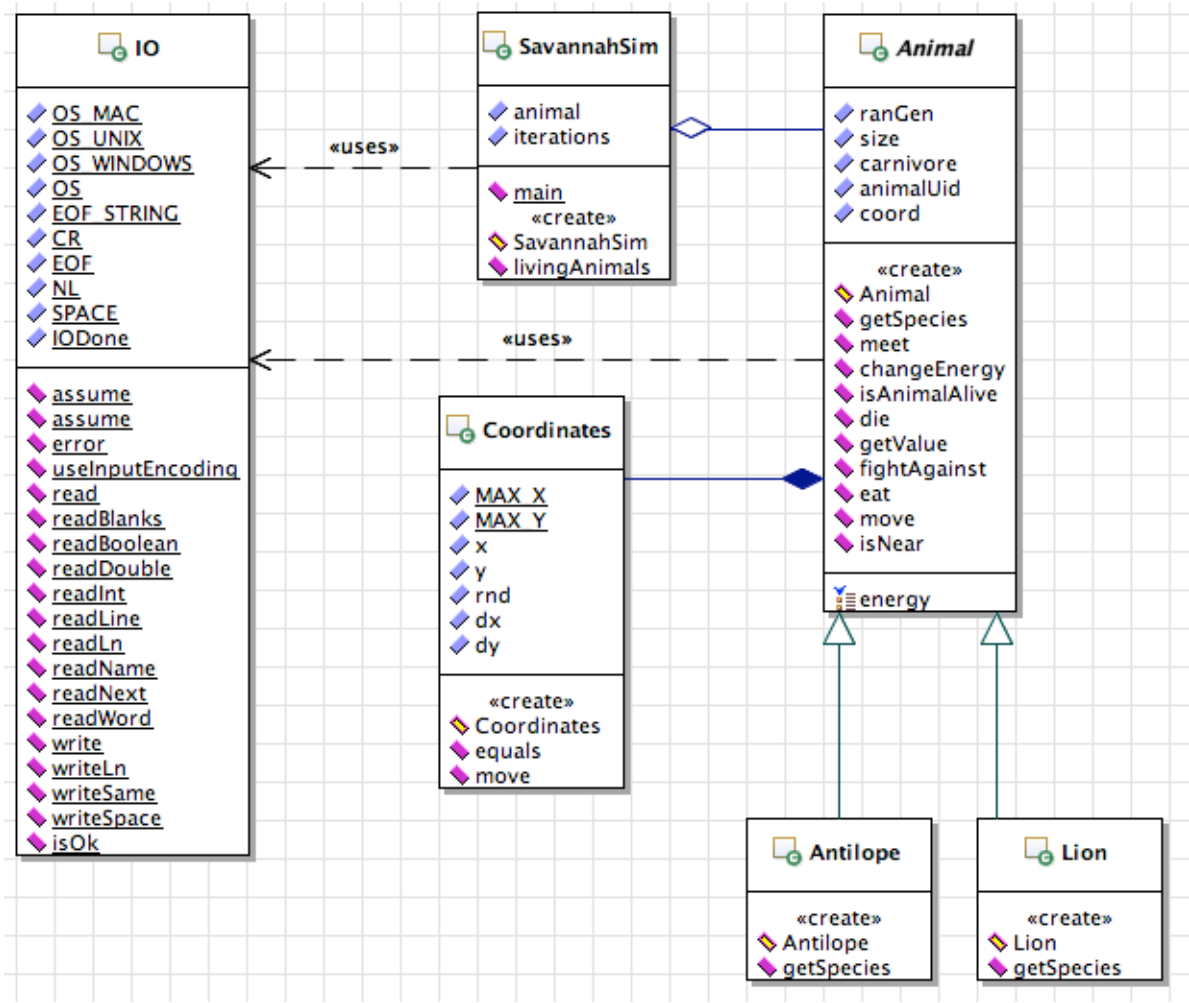
Aufgabe 2.1: Architekturüberblick (1)



- Ziel
  - Grobüberblick über die Organisation der Software
  - Einführungsaufgabe um den Code zu verstehen
- Kriterien
  - Wahl einer geeigneten Notation
  - Sinnvoll modellierte Architektur
    - Fokus auf **Architektur** - Attribute und Operationen von Klassen sind hier weniger relevant ...
  - Beschreibung der Komponenten
- Reverse Engineering Klassendiagramm ≠ idealer Architekturüberblick
  - Fokus auf wichtige Klassen, oder auch ein abstraktes Objektmodell

Aufgabe 2.1: Architekturüberblick (2)

- Reverse Engineered (Borland Together) und *angepasst*:



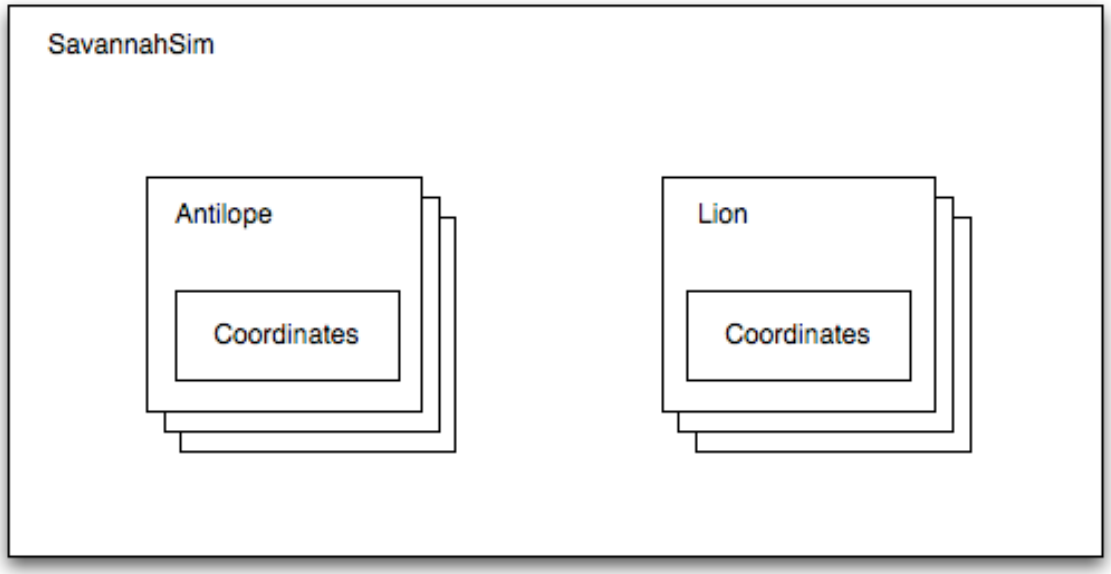
Teil-Ganzes Beziehungen:

- Aggregation
  z.B. Auto ◇---- Rad

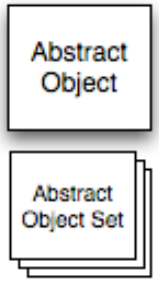
- Komposition (Sonderfall der Aggregation, mit Existenzabhängigkeit)
  z.B. Mensch ◆---- Gehirn

Aufgabe 2.1: Architekturüberblick (3)

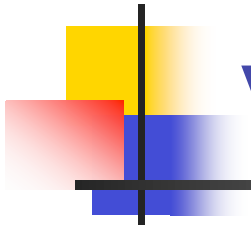
- Abstraktes Objektmodell, *ergänzend* (zeigt die Hierarchien der Komponenten)



Notation:



Aufgabe 2.2: Analyse der verw. Architekturstile (1)



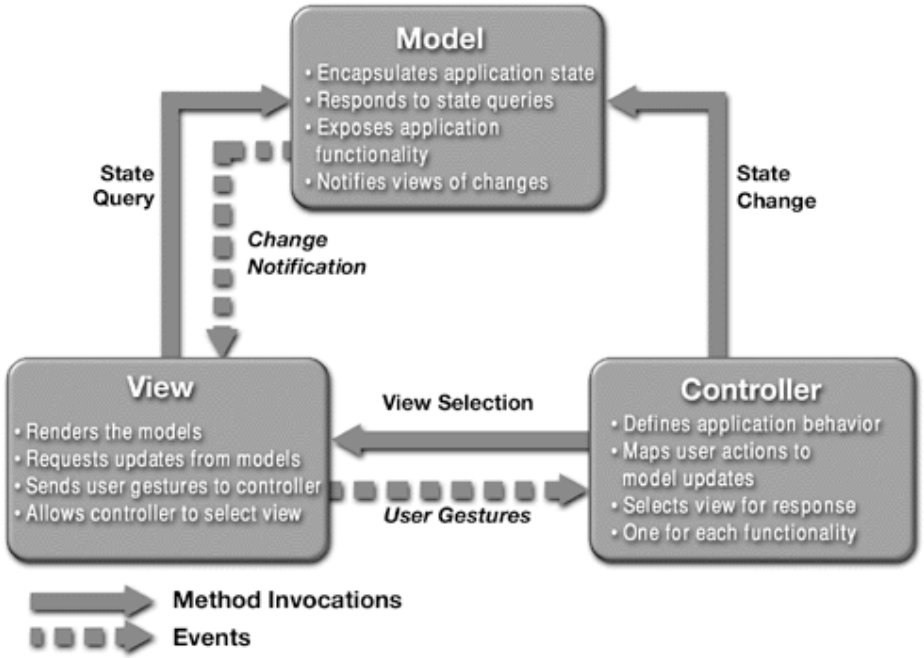
- Ziel
  - Erweiterung der Software um ein GUI (graph. Benutzerschnittstelle)
  - Finden eines passenden Architekturmusters für diese Erweiterung
- Architektur
  - Auf Ebene von Modulen bzw. Komponenten
  - Kooperationsformen zwischen den Modulen
- Architekturstil (≠ Entwurfsmuster!)
  - Vokabular zur Architekturbeschreibung
  - Regeln und Einschränkungen, Semantische Interpretation

Aufgabe 2.2: Analyse der verw. Architekturstile (2)

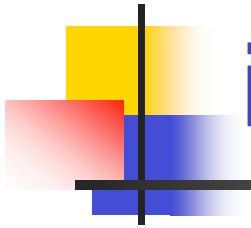
- Analyse der beschriebenen Architekturstile aus der Vorlesung
- Idealer Architekturstil für diese Aufgabe:
  - **Model-View-Controller**

<u>Controller:</u>
- Steuerung Benutzer
- Änderungen
- Propagierung von Änderungen an View

<u>Model -> View und View -> Controller:</u>
- *Observer* Design Pattern



Aufgabe 2.3: Entwurfsmuster in der Software (1)

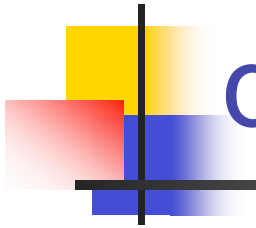


- Ziel
  - Verwendung bewährter, vorgefertigter “Lösungsschablonen” für wiederkehrende Entwurfsprobleme
  - Begriffliche Basis und Terminologie für Entwurfsbeschreibung (Design).

- Passendes Entwurfsmuster zu Model-View-Controller
  - **Beobachter (Observer Pattern)**
  - “Definiere eine 1-zu-n-Abhängigkeit zwischen Objekten, so dass die Änderung des Zustands eines Objekts dazu führt, dass alle abhängigen Objekte benachrichtigt und automatisch aktualisiert werden.”

  - Teilweise gab es (wie zu Aufgabe 2.2) sehr falsche Lösungen ...

Aufgabe 2.4: Dokumentation der (erweiterten) Architektur (1)



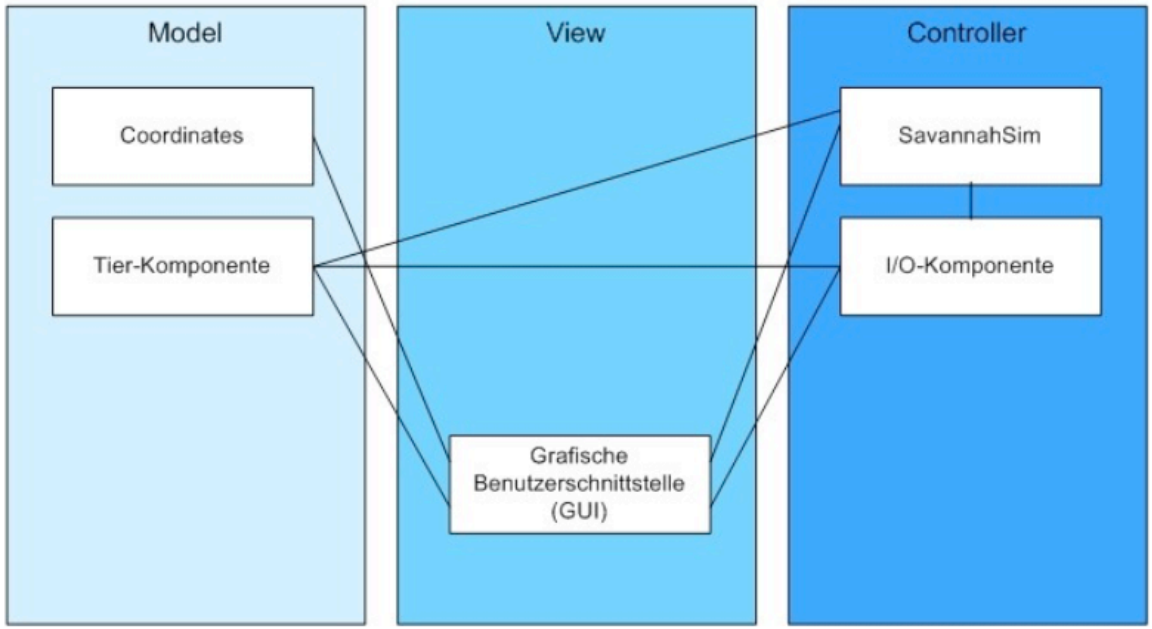
- Ziel
  - Erweiterung der gegebenen Architektur (Aufgabe 2.1) um ein GUI
  - Gemäss des von uns gewählten Architekturstils und Entwurfsmusters zur Realisierung.

- Notation
  - Fokus auf die Architektur (Blick auf das Ganze, keine Implementierungsdetails)
  - Spezifizierung der Beziehungen/Rollen zwischen Systemteilen

Aufgabe 2.4: Dokumentation der (erweiterten) Architektur (2)

- Mögliche Lösung

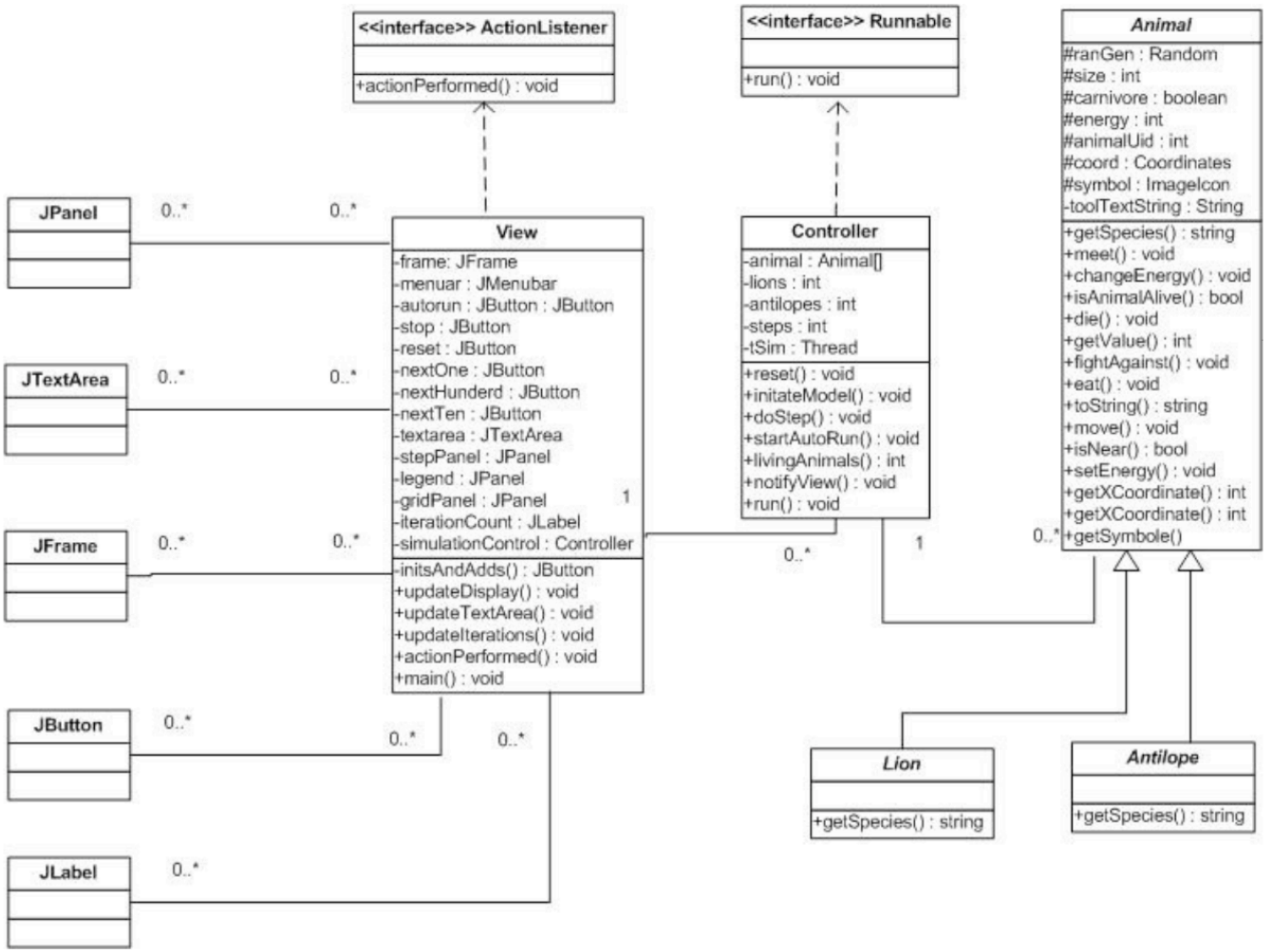
(aus einer Abgabe)



Aufgabe 2.4: Dokumentation der (erweiterten) Architektur (3)

- Mögliche Lösung

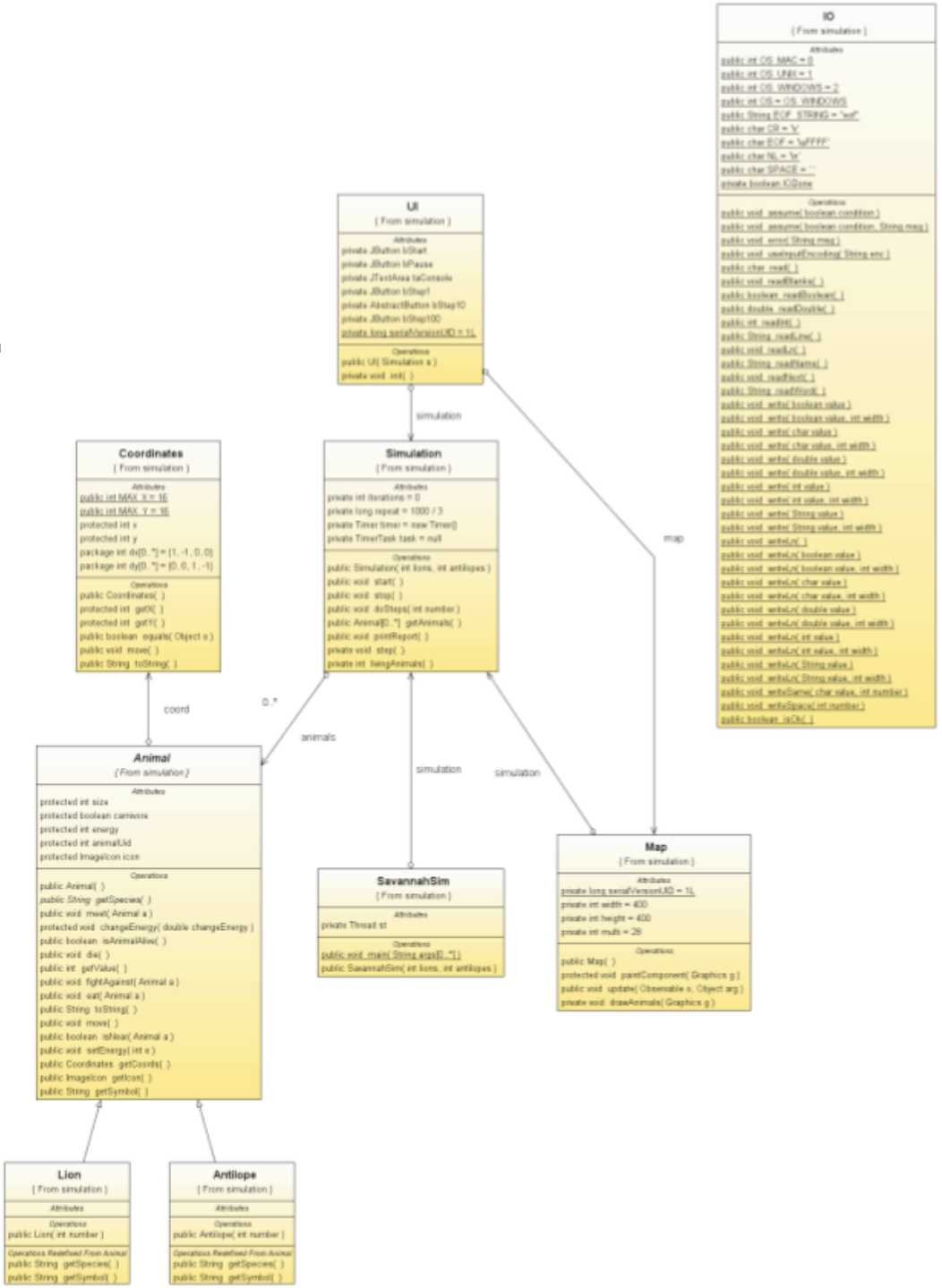
(aus einer Abgabe)



Aufgabe 2.4: Arch. (4)

- Mögliche Lösung

(aus einer Abgabe)



Hier allerdings zu viel Implementierungsdetail (Attribute und Operationen)

Aufgabe 2.5: Architektur-anpassungen, GUI-Implem. (1)

- Ziel
  - Realisierung der Architekturanpassung im Code
  - Implementierung eines passenden, einfachen GUIs
- Die Aufgabe wurde nicht von allen gelöst. Teilweise gab es gute Lösungen!
- Kriterien
  - Überprüfung der geforderten Funktionalität als Black-Box-Test

Aufgabe 2.5: Architektur-anpassungen, GUI-Implem. (2)

- **Demo**
- einiger guter Lösungen ...

Danke für die Aufmerksamkeit.