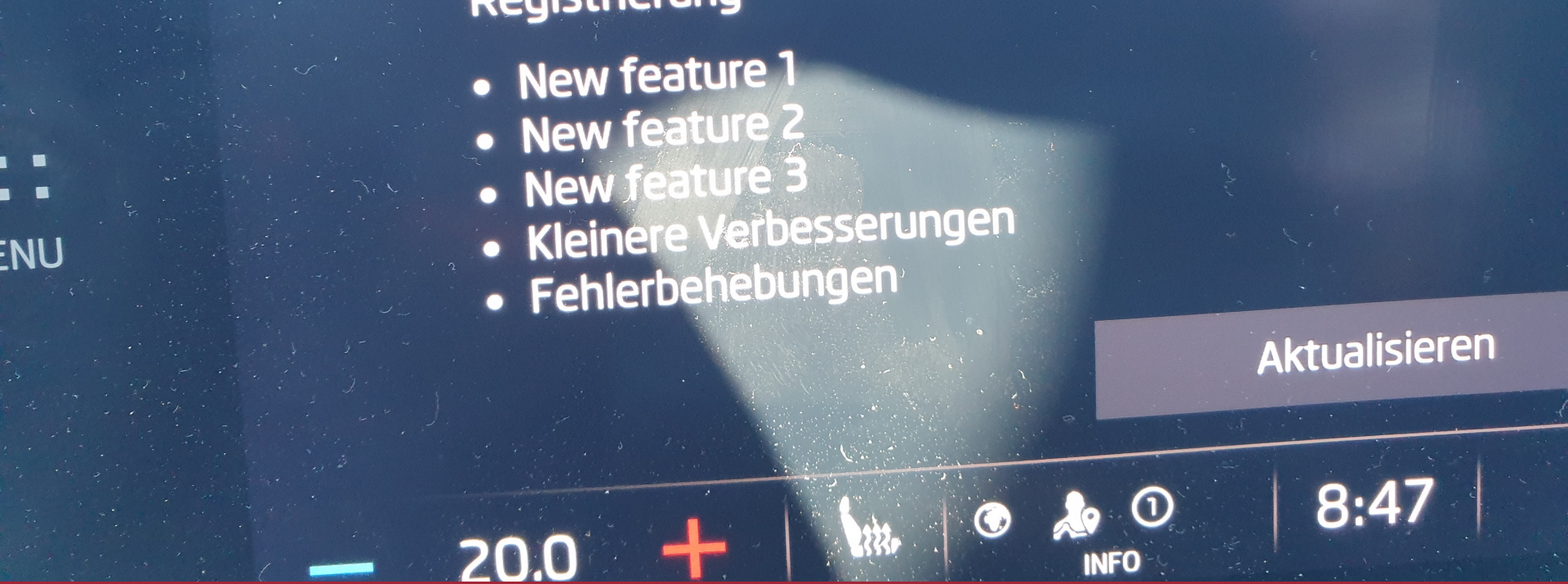


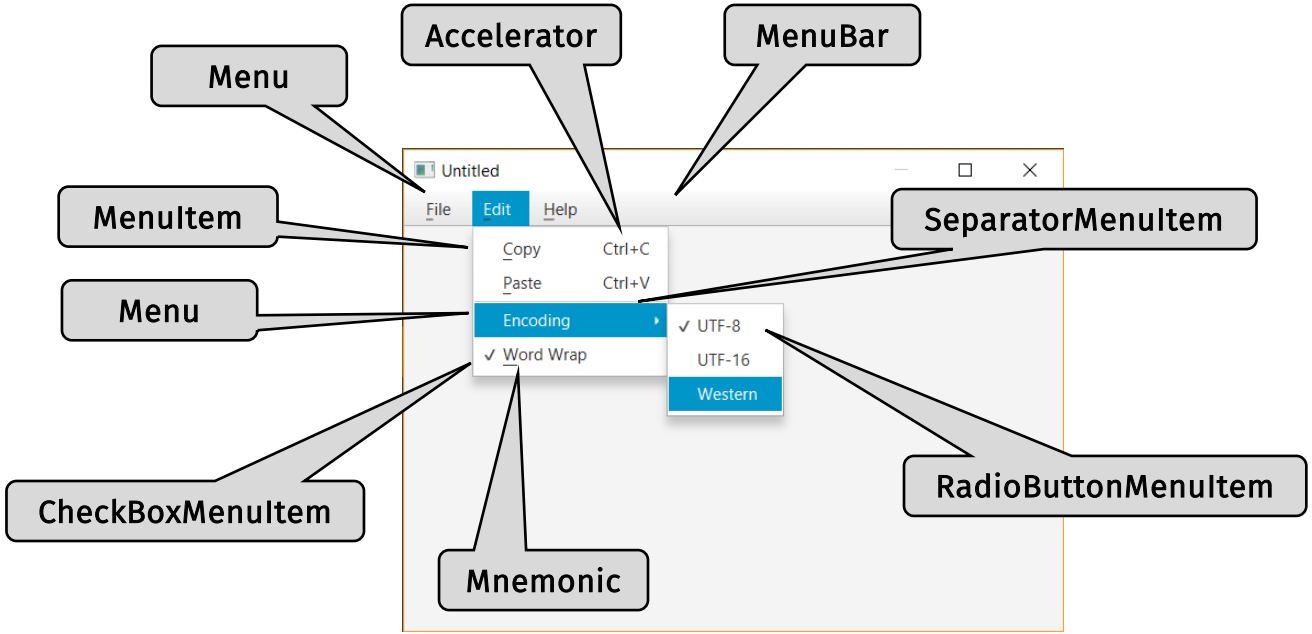
09-GUI-3-MVC

Objektorientierte Programmierung | Matthias Tichy

Lernziele

- Weitere GUI-Elemente
- Zusammenhänge zwischen UI, Daten und Interaktionen

Menü


Accelerator
MenuBar
Menu
MenuItem
SeparatorMenuItem
Menu
CheckBoxMenuItem
Mnemonic
RadioButtonMenuItem
Untitled
File
Edit
Help
Copy Ctrl+C
Paste Ctrl+V
Encoding
Word Wrap
UTF-8
UTF-16
Western

ToolBar

- kann beliebige Komponenten aufnehmen
- Typischerweise mit Icons versehen
- mehrere möglich

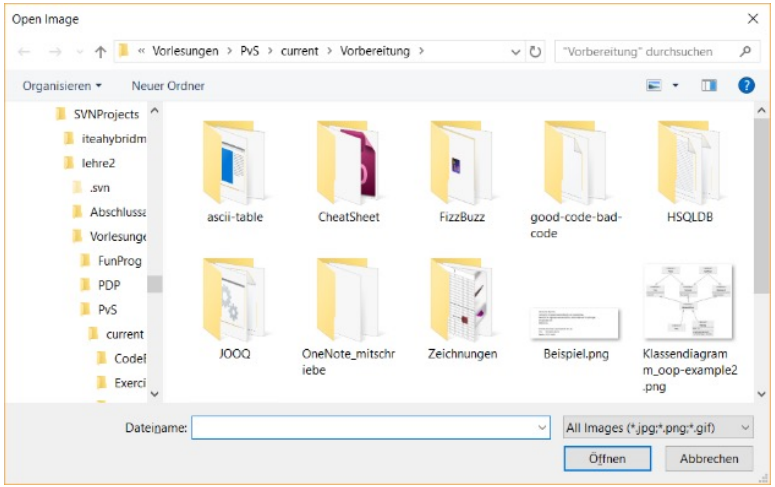
- Oft nur Abkürzung für andere Aktionen → EventHandler wieder verwenden

Dialog

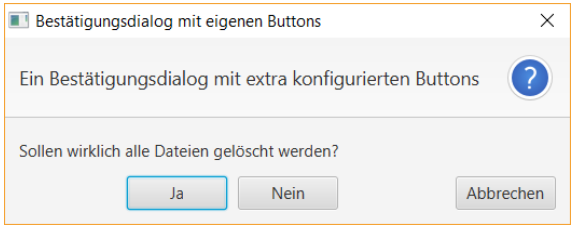
- Fenster, um einen bestimmten Bereich der Eingabe zu gruppieren
- Modal vs. Nicht-Modal
- einige vordefinierte Dialoge
  - FileChooser
  - ColorPicker
  - Alert, TextInputDialog, ChoiceDialog

FileChooser

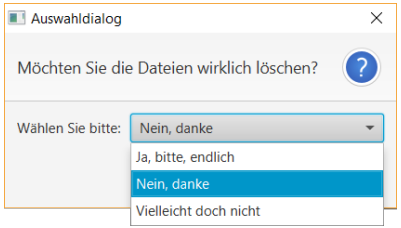
- nativer Dialog wird aufgerufen
- typische Einstellmöglichkeiten:
  - Startverzeichnis
  - Datei-Filter
  - Multiple-Selection erlaubt?
  - nur Ordnerauswahl



Alert, Eingabedialoge

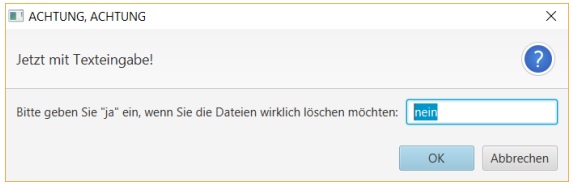


Alert

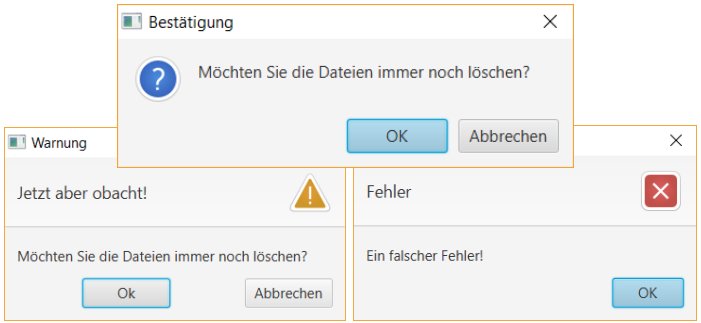


ChoiceDialog

TextInputDialog

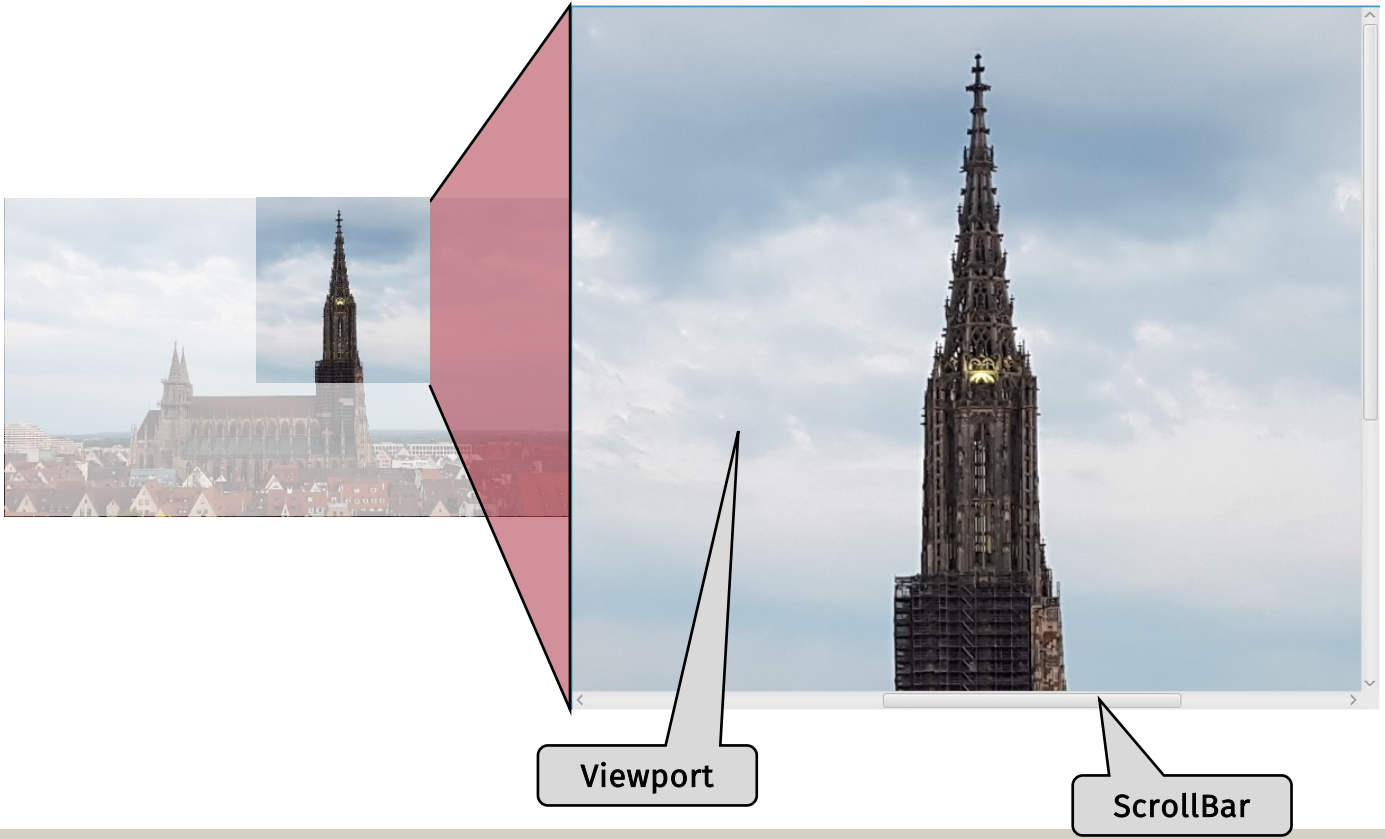


Alert

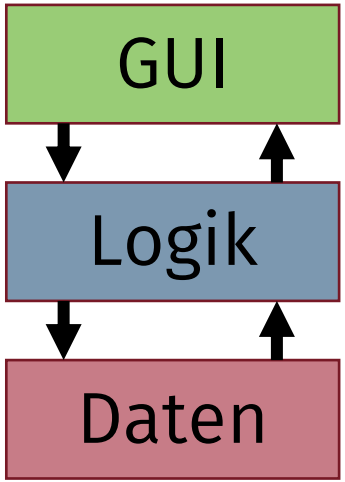


- Optionen:
  - Text, Titel, Typ, Icon
  - speziellere Optionen (Auswahl, Buttonbeschriftung) je nach Art

ScrollPane



UI vs. Daten vs. Interaktionen



Problem

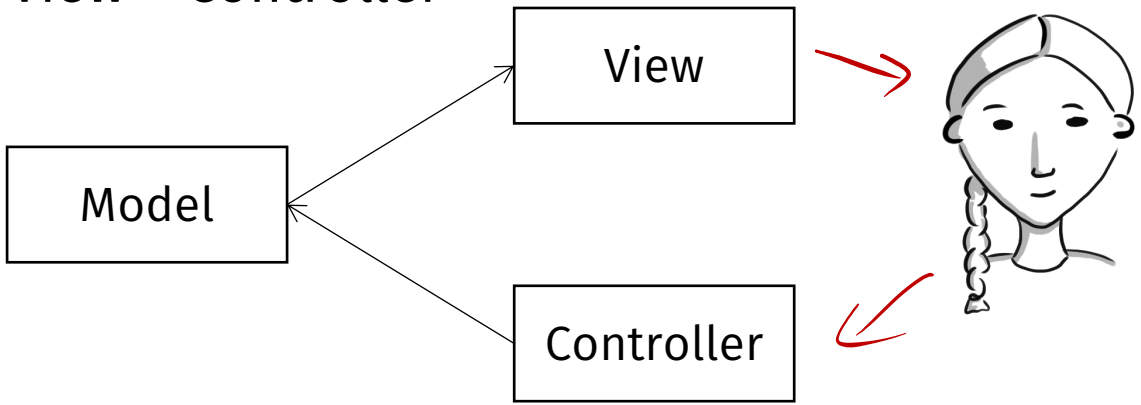
Fragen

- Wie können wir den Code sauber separieren?
- Welche Teile gehören zusammen?
- Wie läuft die Kommunikation zwischen den einzelnen Teilen?

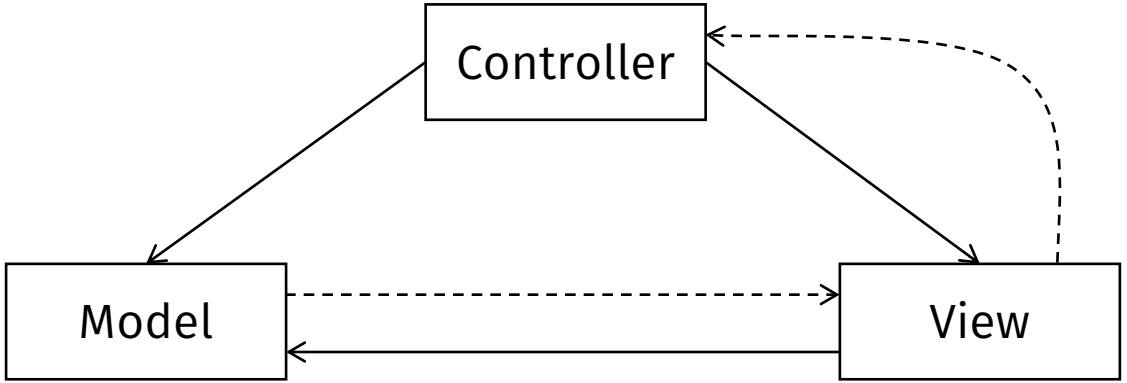
Lösung

- Probleme schon sehr alt:
  1979 mit der ersten grafischen Oberfläche

- Vorschlag von damals:
  Model – View – Controller



Model-View-Controller heute



direkte Verbindung (Referenz)

indirekte Verbindung (z.B. Observer)

Rollen

- Model
  - darzustellende Daten (auch mehrere Klassen)
  - Teil der "Logik" (Anwendungsschicht)
  - "dümmster" Teil (kennt weder View noch Controller)

- View
  - verantwortlich für Darstellung der Daten
  - ermöglicht Interaktion über Ereignisse
  - Dialogablauf unbekannt

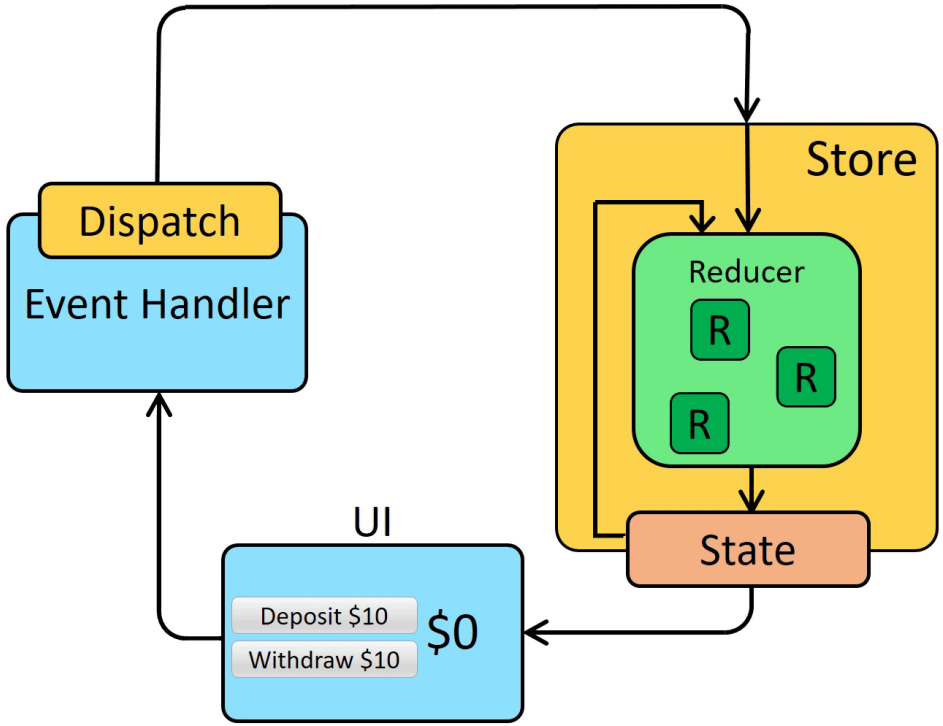
Rollen

- Controller
  - Verbindung zwischen Model und View
  - oft Teil der Präsentationsschicht
  - empfängt Ereignisse der View
  - Umformung zu Daten für Model
  - leitet Ergebnisse zurück an View (evtl. auch direkt vom Model über Observer-Pattern)
  - typischerweise ein Controller pro View

offene Aufgaben

- Geschäftslogik
  - meist extra Klassen
- Validierung von Benutzereingaben
  - oft in der View
- Datenformatierung
  - oft auch in der View
  - extra ViewModel (→ Model - View- ViewModel)

Web: Redux (State / Events / Actions)



https://redux.js.org/tutorials/essentials/part-1-overview-concepts

Lernziele

- Weitere GUI-Elemente
- Zusammenhänge zwischen UI, Daten und Interaktionen