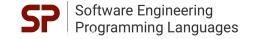


09-GUI-3-MVC

Objektorientierte Programmierung | Matthias Tichy

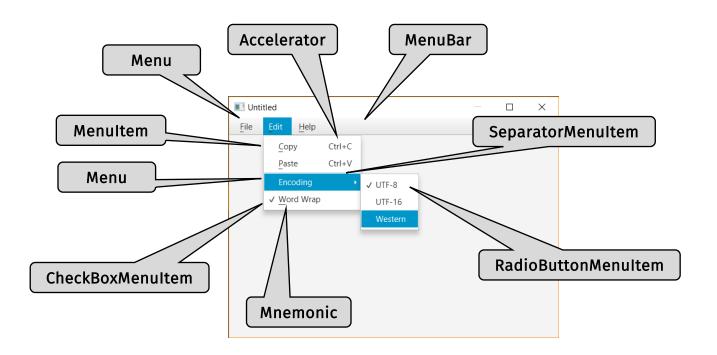




Lernziele

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

Menü



ToolBar

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

■ Oft nur Abkürzung für andere Aktionen → EventHandler wieder verwenden

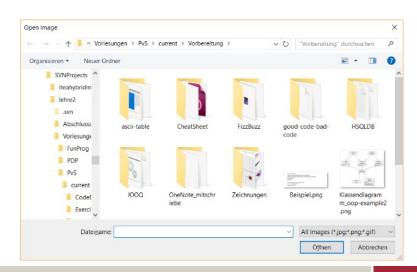
Dialog

- 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



Möchten Sie die Dateien wirklich löschen?

Wählen Sie bitte:

Nein, danke

Ja, bitte, endlich

Nein, danke

Vielleicht doch nicht

ChoiceDialog

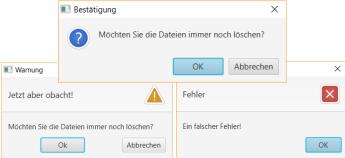
Auswahldialog

TextInputDialog



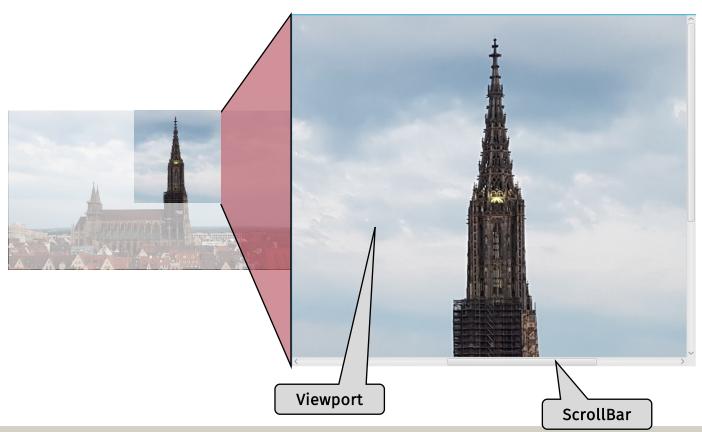


X

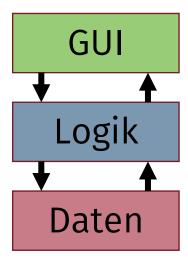


- Optionen:
 - Text, Titel, Typ, Icon
 - speziellere Optionen (Auswahl, Buttonbeschriftung) je nach Art

ScrollPane



UI vs. Daten vs. Interaktionen



Problem

```
mort jerefr.application.Elations;
mort jerefr.event.ActionDeent;
mort jerefr.event.EventHandler;
mort jerefr.gemetry.Eventy;
    part, pavefer, geometry, Ordentation,
     SEEL javain.gomm.gomm;
SEEL javain.gomm.combroil.blast;
SEEL javain.gomm.combroil.blast.blast?ype;
SEEL javain.gomm.combroil.blattom;
     art. javafa.come.control.Lobel.r
     1994. [avefix.queen.cocked.Benzy
1994. [avefix.queen.cocked.Benzikery
1994. [avefix.queen.cocked.Benzikery
     sirt. Tavafa.comm.control.Streel.Para
     idit. javafa. queen. madred. Departmenter Den Der
1986. javafa. queen. madred. Qafdidiran y
1986. javafa. queen. madred. Vestifera y
     sirt. Tavafe, come, control. Vest (rest)
    sart 'avafe, come, control, Tooltary
   pirt. Javafe. prese. Jeage. Scape;
pirt. javafe. prese. Jeage. Scapelifer;
pirt. javafe. prese. Jegut. ReyCoskication
     sigh, Yawafin, powers, Japanah, Reschoelbarer
 more javels.gome.lagest.Confecture;
aport javels.gome.lagest.Confr
aport javels.gtage.Slotheaper;
aport javels.gtage.Sloge;
import jave.Sc.1685e;
import jave.Sc.1686ceptSen;
import jave.net.MedScneed1886ceptSen;
import jave.net.1886;
    port jave.util.Optionel;
    blic class (Winegelieser attanda Application implementa EventHandler-Schientbeett (
 public static vold male@bring[] orgo) {
    learnh (orgo) r
   public void phort(Shape primary(hape) thises Exception (
this primary(hape = primary(hape)
Secular member = the Secular();
```

```
menalida.gehthem; ().middli (menalhenicasi, menalhenipentri, menalheniawa, paparahar,
Herealtern were iterritiffe = this Herealtern\{i_n\}_{n=1}^{\infty} were iterritiffe, get Armeler stor (KeyCochination, volueCff were iterritiffe);
Toolfar boolfar - der Toolfarfir
      ilitEann oplitEann = now SplitEann);
  tabel liditoprolytion = nor tabel ("Reprolytion")
                  dienting gelfalisiere j. mitali (intelligencephion, beliegeregibion)
mm., mit gelfanden (libitageregibion), EREND (mitali gelfali);
mm., gelfali dienther (libitageregibion), EREND (mitali gelfali);
mm., gelfali dienther (libitageregibion), EREND (mitali gelfali);
mm., gelfali dienther (kalbageregibion), EREND (mitali gelfali);
mm., gelfali dienther (kalbageregibion), EREND (mitali gelfali);
mm., gelfali dienther (kalbageregibion), EREND (mitali gelfali);
```

```
nchowFane, setTopAnchor (1b1FileName, BORDER WIDTH) /
 InchorPane, setLeftEnchorflblFileName, BORDER WIDTH:
 AnchorPane.setTopAnchor(imageScrollPane, BORDER WIDTH * BORDER WIDTH)
 imageFane.getChildren().add&ll(lhlFileName.imageScrollFane)
 splitFame.getItems()_addAll(imageFame, descriptionFame);
primaryStage.setScene(scene);
primaryStage_show();
 * Opens a FileChnoser and displays the image in the GUL. */
Mwarrida
FileChonser filedialog = new FileChonser();
 filedialog.setInitialDirectory(new File(System.getProperty("user_home"))
 filedialog.getExtensionFilters().addMl1(
   new FileChooser.ExtensionFilter("N11 Nagges", "*.jgg", "*.gag", "*.gag"),
     new FileChooser.ExtensionFilter("JEG", "".jgg"],
   new FileChanger.ExtensionFilter("FHGF, "0.3225"),
     new FileChooser, ExtensionFilter ("SIF", "", gif"]];
 File result = filedialog.showlponDialog(primaryStage)
 12 (result to cull) :
```

Fragen

• Wie können wir den Code sauber separieren?

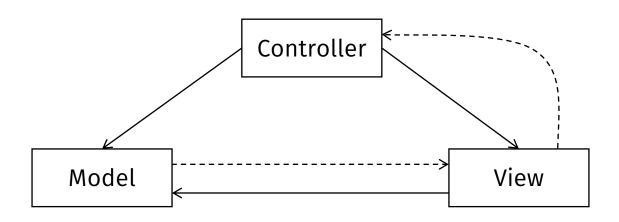
- Welche Teile gehören zusammen?
- Wie läuft die Kommunikation zwischen den einzelnen Teilen?

Lösung

Probleme schon sehr alt:
 1979 mit der ersten grafischen Oberfläche

Vorschlag von damals:
 Model – View – Controller
 Model
 Controller

Model-View-Controller heute



direkte Verbindung (Referenz)

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

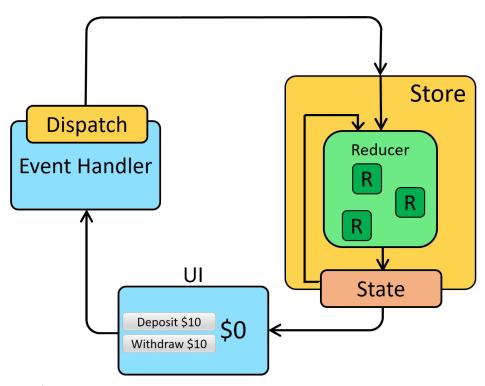
- Controller
 - Verbindung zwischen Model und View
 - oft Teil der Präsentationsschicht
 - empfängt Ereignisse der View
 - Umformung zu Daten für Model
 - leitet Ergebnisse zurück an View (evtl. auch direkt vom Model über Observer-Pattern)
 - typischerweise ein Controller pro View

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