

GUI Komponenten Komponenten und Wiederverwendung

Sommersemester 2024 Stefan Langer Veronika Dashuber

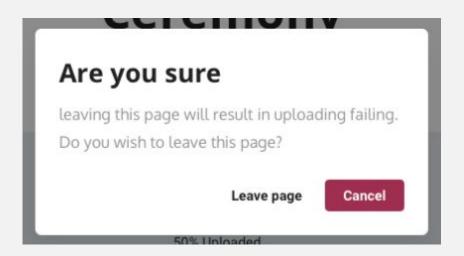
Agenda

- 1. Komponenten, Schachtelung & Wiederverwendung
- 2. Demo "Main- & Form-View in JavaFX"
- 3. GUI Komponenten mit JavaFX

Was ist eine GUI Komponente?



- Voll funktionsfähiger und wiederverwendbarer Baustein
- Kann in verschiedenen Kontexten eingesetzt werden
- Basiert ggf. selbst auf wiederverwendbaren Bausteinen
- Beispiele
 - Ein einfaches Steuerelement (z.B. Button)
 - Ein zusammengesetztes Steuerelement (z.B. Datumsauswahl)
 - Ein Dialog-Bereich zur Ansicht und Eingabe von Adressdaten
 - Ein kompletter Dialog
 - Eine Abfolge von Dialogen



Eigenschaften von GUI Komponenten

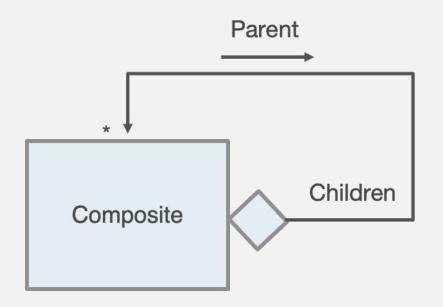


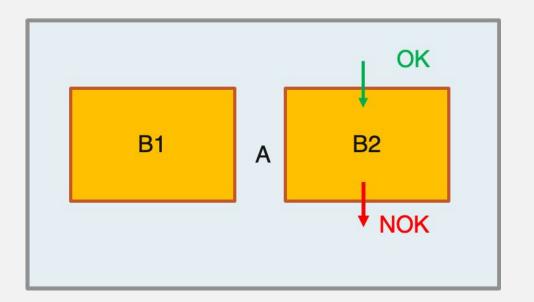
- Mehrfach instanziierbar
 - Jede Komponente benötigt eigene Daten (Model, Presenter, View)
- Eventbasiert
 - O Komponenten benachrichtigen ihre Nutzer mittels Event-Schnittstellen
- Wiederverwendbar
 - Eine Komponente kann in beliebigen Dialogen eingesetzt werden
- Kontextfrei
 - Eine Komponente darf nicht das Elternelement bzw. den umgebenden Kontext kennen.

Kontextfreie Komponenten



Damit eine Komponente wiederverwendbar ist darf sie nicht direkt den Parent aufrufen.

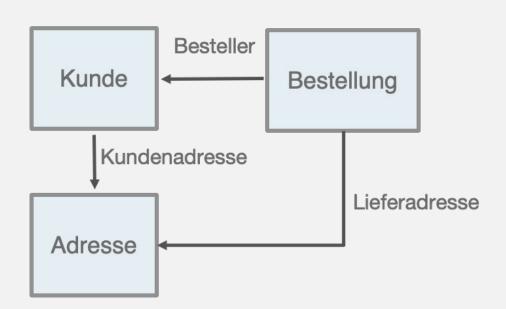


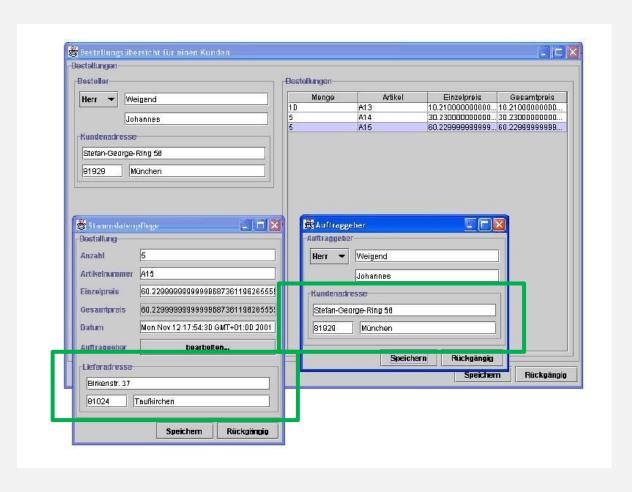


ABER: Die Kommunikation von B zu A ist zum Beispiel über Listener möglich.

Beispiel: Wiederverwendung der Adresse in einer Bestellung



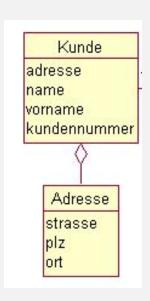


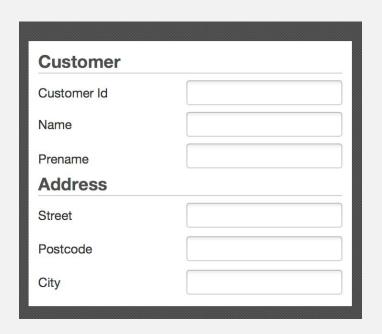


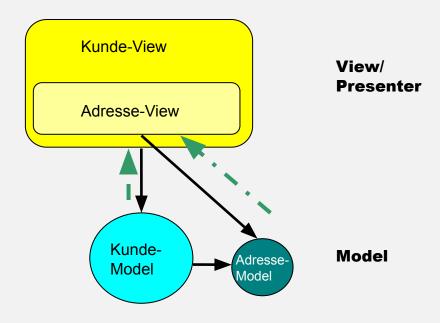
MV* bei GUI Komponenten



Bei zusammengesetzten GUI Komponenten sind auch die Modelle zusammengesetzt







Das Presentation Model Pattern



Eine direkte Verwendung von Business-Objekten als UI Modelle ist nicht ratsam

Unabhängiges UI-Modell:

Das Presentation Model ist vom Business-Modell getrennt und repräsentiert die Zustände und Verhalten der Benutzeroberfläche, nicht die Geschäftslogik.

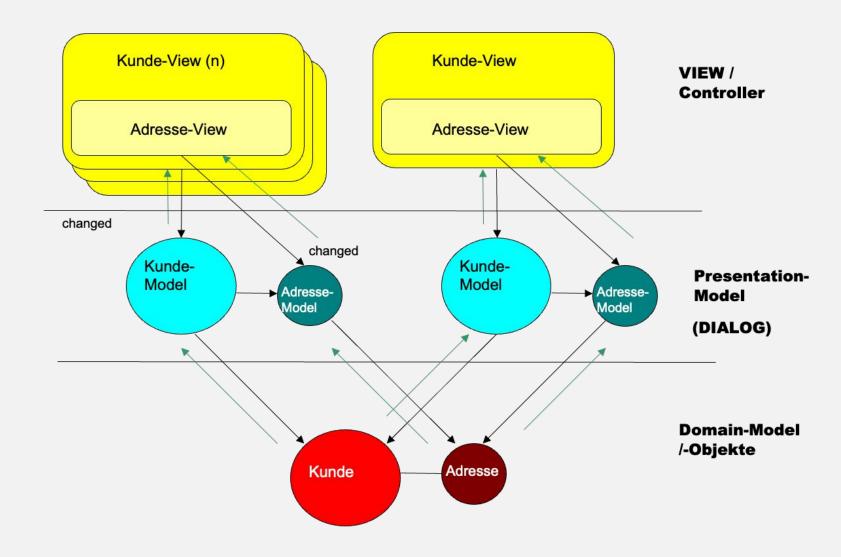
Spezifische UI-Zustände:

- Das Presentation Model enthält UI-spezifische Zustände wie die Auswahlzustände von Kontrollelementen (z.B., ob eine Checkbox ausgewählt ist), die in der Geschäftslogik nicht verwendet werden.
- Inkonsistenzmanagement: Das Presentation Model kann zeitweise von den Business-Modellen abweichen, etwa während der Validierung von Eingaben (z.B., wenn eine E-Mail-Adresse noch nicht validiert ist), während die Business-Objekte immer konsistent bleiben.

http://martinfowler.com/eaaDev/PresentationModel.html

Präsentation-Model vs. Domain-Model



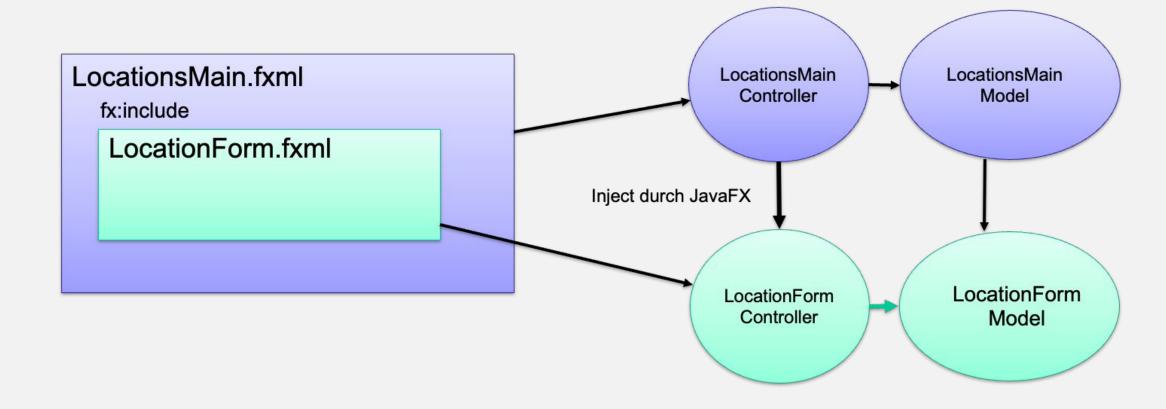


Agenda

- 1. Komponenten, Schachtelung & Wiederverwendung
- 2. Demo "Main- & Form-View in JavaFX"
- 3. GUI Komponenten mit JavaFX

Demo: Schachtelung von Oberflächen





Agenda

- 1. Komponenten, Schachtelung & Wiederverwendung
- 2. Demo "Main- & Form-View in JavaFX"
- 3. GUI Komponenten mit JavaFX

GUI Komponenten mit JavaFX



- Komponenten bestehen aus
 - View (FXML),
 - Controller/ Presenter und
 - Model.
- Komponenten lassen sich mit JAVA Code und/oder FXML bauen.

<fx:include>

The <fx:include> tag creates an object from FXML markup defined in another file. It is used as follows:

```
<fx:include source="filename"/>
```

where *filename* is the name of the FXML file to include. Values that begin with a leading slash character are treated as relative to the classpath. Values with no leading slash are considered relative to the path of the current document.

For example, given the following markup:

If my button.fxml contains the following:

```
<?import javafx.scene.control.*?>
<Button text="My Button"/>
```

the resulting scene graph would contain a VBox as a root object with a single Button as a child node.

Note the use of the "fx" namespace prefix. This is a reserved prefix that defines a number of elements and attributes that are used for internal processing of an FXML source file. It is generally declared on the root element of a FXML document. Other features provided by the "fx" namespace are described in the following sections.

<fx:include> also supports attributes for specifying the name of the resource bundle that should be used to localize the included content, as well as the character set used to encode the source file. Resource resolution is discussed in a later section.

Komponenten mit FXML



- Über <fx:include> lassen sich andere FXML-Dateien inkludieren.
- Die Inhalte der Kind-Komponente werden an der entsprechenden Stelle eingefügt.

⇒ Wie kann die einbettende Komponente mit der Kind-Komponente interagieren?

Nested Controllers

Controller instances for nested FXML documents loaded via the <fx:include> element are mapped directly to member fields of the including controller. This allows a developer to easily access functionality defined by an include (such as a dialog window presented by an application's main window controller). For example, given the following code:

main_window_content.fxml

```
<VBox fx:controller="com.foo.MainController">
    <fx:include fx:id="dialog" source="dialog.fxml"/>
    ...
</VBox>
```

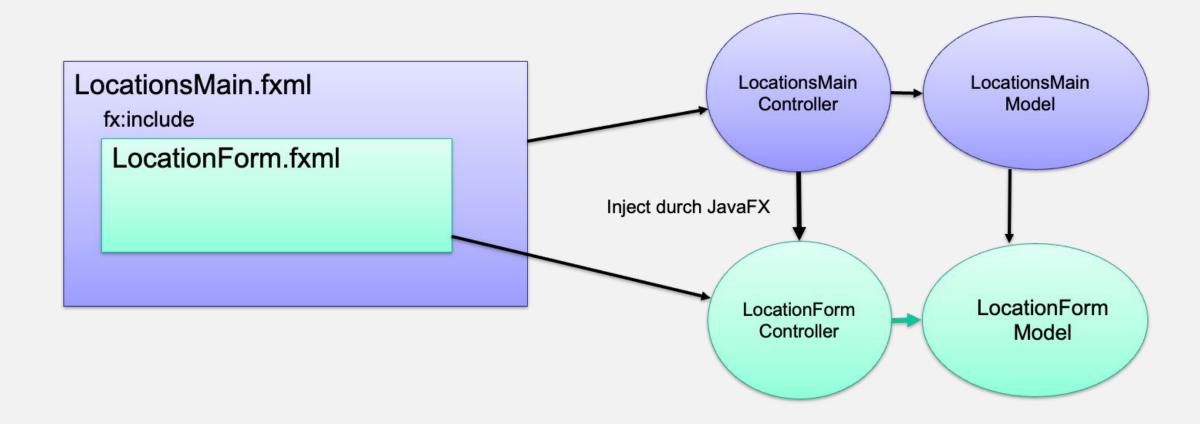
MainController.java

```
public class MainController extends Controller {
    @FXML private Window dialog;
    @FXML private DialogController dialogController;
    ...
}
```

when the controller's initialize() method is called, the dialog field will contain the root element loaded from the "dialog.fxml" include, and the dialogController field will contain the include's controller. The main controller can then invoke methods on the included controller, to populate and show the dialog, for example.

Der Kind-Controller kann injiziert werden (nicht umgekehrt!)





Zusammenfassung



Definition einer GUI-Komponente:

Ein voll funktionsfähiger und wiederverwendbarer Baustein für verschiedene Einsatzbereiche.

■ Eigenschaften von GUI-Komponenten:

Mehrfach instanziierbar, eventbasiert, wiederverwendbar und kontextfrei.

MV bei GUI-Komponenten*:

Modelle sind oft zusammengesetzt, umfassen separate Ansichten und Datenlogik.

Presentation Model Pattern:

Trennt UI-Modelle von Geschäftslogik zur besseren Verwaltung von UI-Zuständen und Validierungen.

■ Komponentenstruktur mit JavaFX:

Nutzung von View (FXML), Controller/Presenter und Model, unterstützt durch das <fx:include>-Tag zur Modulintegration.

Welche Fragen habt ihr?