
GUI Vorlesung 2016

Übung 12 - Prüfungsvorbereitung

Johannes Weigend - 27.06.2016

Vorlesung 1 - Grundlagen grafischer Oberflächen

- 1) Nennen Sie wichtige Einflussfaktoren für den Entwurf grafischer Oberflächen.
Siehe Folie 17
- 2) Grenzen Sie ab: FAT-Client, Thin-Client, Rich-Client
Siehe Folie 19
- 3) Nennen Sie drei Beispiele für deklarative Oberflächen
Siehe Folie 31
- 4) Gegeben ist eine JavaFX UI-Komponente SearchView (Basistyp Parent). Erstellen Sie ein Anwendung die diese Komponente anzeigt (Kein FXML)
Siehe Folie 33
- 5) Was muss an der Anwendung geändert werden, wenn anstelle der UI-Komponente eine FXML Datei geladen werden soll?
Siehe Folie 34
- 6) Was machen die git Kommandos clone und pull?

Vorlesung 2 - Gestaltungsregeln

- 1) Was versteht man unter Response Design?
Siehe Folie 14
- 2) Warum macht eine Erstellung von Mock-Ups Sinn bei der Erstellung grafischer Oberflächen?
Siehe Folie 18

3) Welche Technologien gibt es für die Entwicklung grafischer Benutzeroberflächen mit Java?

Siehe Folie 25

4) Zeigen Sie grafisch den Aufbau einer JavaFX Anwendung.

Siehe Folie 37

5) Erklären Sie das Composite Entwurfsmuster anhand eines UML Diagramms.

Siehe Folie 38

6) An welcher Stelle der JavaFX Klassenhierarchie befindet sich ein Composite Muster?

Siehe Folie 39

7) An welcher Stelle der JavaFX Klassenhierarchie befindet sich ein Composite Muster?

Siehe Folie 39

8) Welche Layout-Panes gibt es in JavaFX. Erklären Sie kurz die Funktion.

Siehe Folie 41

Vorlesung 3 - Events

1) Warum braucht jede grafische Oberfläche einen Event-Loop?

Siehe Folie 4

2) Warum ist es sinnvoll sowohl Event-Bubbling als auch Event-Capturing zu unterstützen?

Siehe Folie 6

3) Warum gibt es bei JavaFX ein Interface „Initializable“. Kann man den Code nicht auch in den Konstruktor verlagern?

Siehe Folie 12

4) Wie lassen sich bei FXML Eventhandler deklarativ mit dem Controller verbinden?

Siehe Folie 16

5) Was macht die Methode „addEventFilter(EventHandler)“

Siehe Folie 17

6) Wie kann man Events in einem EventHandler filtern?

event.consume();

- 7) Warum gibt es eine Klassenhierarchie von Eventtypen bei JavaFX? (z.B. Event.ANY -> InputEvent.ANY -> MouseEvent.ANY ...). Zeigen Sie ein Beispiel.

node.addEventFilter(Event.ANY, e -> e.consume()); // Blockiert alles

- 8) Welche Möglichkeiten gibt es EventHandler und Controller in Klassen aufzuteilen. Nennen Sie Vor- und Nachteile.

Vorlesung 4 - MV*

- 1) Erklären Sie an einem UML Diagramm das Observer Muster?

Siehe Folie 4

- 2) Beim klassischen MVC Muster: Wo ist das Observer-Muster?

Zwischen dem Model (= Subject) und dem View (= Observer)

- 3) Zeigen Sie anhand einer Skizze die Architektur von MVP mit Databinding. Wenn bidirektionales Binding benutzt wird, wo ist das Observer-Muster?

Sowohl Model ist Subject als auch das UI-Element. Ebenso ist das UI-Element

Observer als auch das Model. Bei Änderungen auf beiden Seiten wird die jeweils andere Seite aktualisiert.

- 4) Bei JavaFX benötigt Databinding sogenannte JavaFX Properties im Model. Programmieren Sie eine Model-Klasse mit einem Integer - Attribut mit dem Namen „count“.

Siehe Folie 14

- 5) Die grafische Oberfläche hat einen Slider. Slider besitzen eine Methode `valueProperty()`. Zeigen Sie den notwendigen Code in der `initialize()` Methode des Controllers mit der beide Properties bidirektional verbunden werden.

slider.valueProperty().bindBidirectional(model.countProperty()).

Vorlesung 5 - GUI Komponenten

- 1) Nennen Sie vier Eigenschaften von GUI Komponenten.

Siehe Folie 4

- 2) Erklären Sie das Presentation Model Muster?

Siehe Folie 7

- 3) Wofür braucht man die FXML-Angabe <fx:include ...>?

Siehe Folie 11

- 4) Wie schachtelt man Controller? Wofür braucht man das?

Wird bei JavaFX in FXML eine Komponente per <fx:include id=„xyz“>

eingebunden, kann man im Controller sich den Controller injizieren lassen. Die Voraussetzung dafür ist das es ein Attribut im Controller mit dem Namen xyzController gibt (wenn privat dann mit @FXML annotiert). Der innere Controller wird benötigt um Eventhandler installieren zu können und das innere Modell initialisieren zu können.

Vorlesung 6 - Parallelität

- 1) JavaFX hat eine Klasse Task zur Vereinfachung der Parallelprogrammierung. Was macht die Klasse?

Siehe Folie 4-9

- 2) In einer Anwendung wird ein Task erzeugt. Dieser überschreibt die Methoden call, succeeded, failed. In welcher der Methoden ist ein direkter Zugriff auf die Oberfläche erlaubt?

Sowohl succeeded() als auch failed() laufen im UI Thread, ein Zugriff auf die Oberfläche ist direkt möglich. call() läuft asynchron. Ein Zugriff ist nur per Platform.invokeLater() erlaubt.

- 3) Wie unterscheiden sich die Klassen Task und Service?

Service kapselt zusätzlich die Threaderzeugung und die Taskerzeugung. Services können wiederverwendet (d.h. mehrfach gestartet werden). Tasks müssen pro Lauf neu erzeugt

werden. Ist der Task fertig (d.h. die Call-Methode wurde verlassen) ist dieser nicht mehr verwendbar.

Vorlesung 8 - Publish Subscribe

- 1) Welche Vorteile bietet ein Eventbus gegenüber einer direkten Kopplung?

Siehe Folie 8

- 2) Erklären Sie das Presentation Model Muster?

Siehe Folie 7

- 3) Erklären Sie das Publish-Subscribe Muster an einem UML Diagramm.

Siehe Folie 10

- 4) Entwerfen Sie Schnittstellen für einen Eventbus

Siehe Folie 13-14

Vorlesung 9 - Architektur

- 1) Was ist eine strikte Schichtenarchitektur. Welche Vorteile bietet diese?

Siehe Folie 2+5

- 2) Welche legalen und illegalen Zugriffe existieren bei Software Komponenten?

Siehe Folie 15+16

Vorlesung 10 - Web UI

- 1) Wie baut man einen View in AngularJS?

Ein View ist eine HTML Datei mit entsprechenden Platzhaltern für die dynamischen Inhalte z.B. {{customer.name}}

- 2) Wo ist das Modell bei AngularJS?

Das Modell ist generisch. Mittels der automatischen Variable \$scope die in den Controller Methoden verwendet werden kann ist ein Zugriff auf Attribute möglich.

- 3) Wo liegt die Dialogsteuerung bei einer AngularJS Anwendung?

Im JS Client per Routendefinition. Der Server ist daran nicht beteiligt. Eine AngularJS Anwendung wird komplett im Browser lokal ausgeführt.

Vorlesung 11 - Migration

- 1) Erklären Sie das Muster „Abstract Factory“

Folie 16 + 17

- 2) Welche Migrationsvarianten zur Migration von Java Swing auf JavaFX gibt es:

Folie 9