UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Software Entwicklung & Programmierung

GUI-Programmierung mit JavaFX 11 (OpenJFX)

PALUNO
The Ruhr Institute for Software Technology

Disclaimer

Bilder und Texte der Veranstaltungsfolien und -unterlagen sowie das gesprochene Wort innerhalb der Veranstaltung und Lehr-Lern-Videos dienen allein dem Selbst- bzw. Gruppenstudium. Jede weiterführende Nutzung ist den Teilnehmenden der Moodle-Kurse untersagt, z.B. Verbreitung an andere Studierende, in sozialen Netzwerken, dem Internet!

Darüber hinaus ist ein studentischer Mitschnitt von Webkonferenzen im Rahmen der Lehre nicht erlaubt.

Agenda

1. **Grundlagen**
2. Komponenten
3. Scene Builder
4. Benutzereingaben/Action Events
5. Anwendung nach dem MVC-Prinzip

Grundlagen

- "**OpenJFX** is an **open source**, next generation client application platform for desktop, mobile and embedded systems built on **Java**."

  https://github.com/openjfx/openjfx.github.io

- In Java eingebundene Frameworks
  - Wurde 2008 erstmals in der Version 1.0 veröffentlicht
  - Löste 2014 SWING als **de facto GUI-Standard** in Java ab
  - Bietet Möglichkeiten, um **Animationen** in Java zu realisieren
  - Liefert einfache Möglichkeiten, um unter anderem Diagramme in Java grafisch darzustellen

Grundlagen

- Es **erleichtert** das **graphische Programmieren** durch vorhandene graphische Komponenten
  - Fenster
  - Menüs
  - Schaltflächen
  - Grafikeinbindung
  - Animationen
  - Abfangen von Events
  - etc.

Grundlagen
Getting Started

https://openjfx.io/openjfx-docs/

- Empfehlung: Run Hello World via Maven oder Gradle
  - Generelle Empfehlung: Benutzt Maven oder Gradle

Beispiel unter:

- https://github.com/openjfx/samples/tree/master/HelloFX

Build Management Tools

Maven und Gradle

- Build Management Tools:
  - helfen bei der Verwaltung des Projekts im kompletten Lebenszyklus (Kompilieren, Testen, Verpacken)
  - Ermöglichen die automatische Einbindung von benötigten Bibliotheken

- Support durch Entwicklungsumgebung:
  - Projekte können direkt als Maven- oder Gradle-Projekt erstellt werden
  - Bietet Projektstruktur gemäß des Build Management Tools inklusive zugehöriger Dateien (z.B. pom.xml)

Mehr zu Maven und Gradle im Foliensatz: Build Management Tools

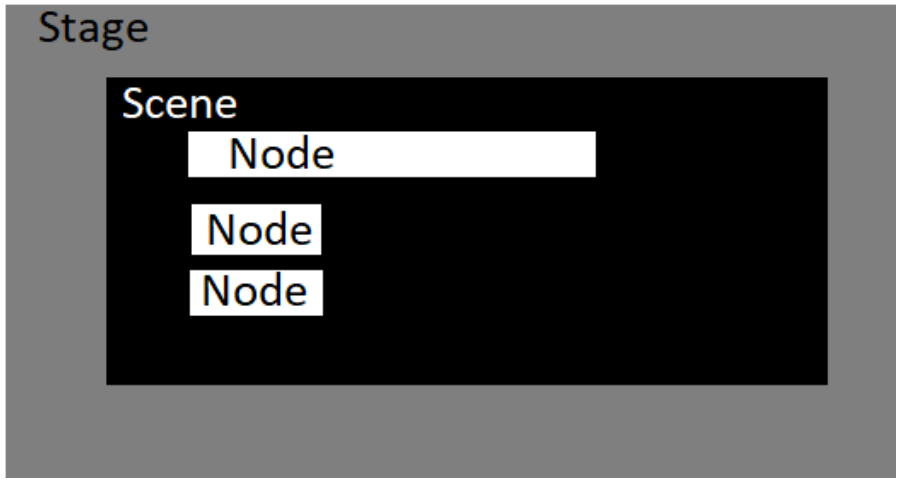
Links: https://maven.apache.org/, https://gradle.org/

Agenda

1. Grundlagen
2. **Komponenten**
3. Scene Builder
4. Benutzereingaben/Action Events
5. Anwendung nach dem MVC-Prinzip

JavaFX - Komponenten

- JavaFX bietet **vorgefertigte Klassen** für die gängigsten graphischen Komponenten an
  - **Stage** für Fenster
    - Stellt die „Bühne“ dar, auf der die GUI präsentiert wird
  - **Scene** für eine leere Fläche in einem Fenster
    - Stellt eine „Szene“ auf einer Bühne dar
  - Dieser „Szene“ können zahlreiche Elemente angehangen werden wie z.B. die Klassen:
    - Button für Buttons
    - Label für Beschriftungen
    - TextField für Textfelder
    - Hyperlink für Links
    - etc.



JavaFX – Fenster öffnen

- Um ein neues Fenster zu öffnen, muss nur die start-Methode der Klasse Application überschrieben und die übergebene primaryStage sichtbar gemacht werden.

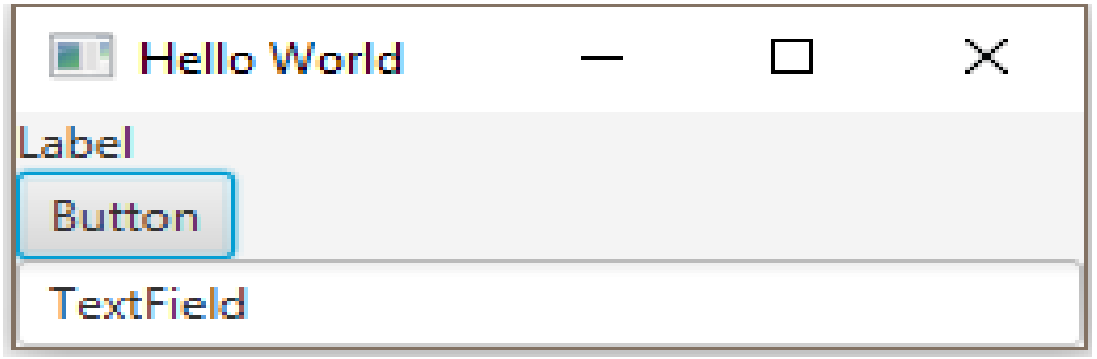
```
public void start(Stage primaryStage) throws Exception {
    primaryStage.setTitle("Hello World");
    primaryStage.show();
}
```

- Das entstandene Fenster
  - hat den Titel „Hello World“
  - kann frei verschoben werden

JavaFX – Fenster öffnen

- **Node** ist Basisklasse für alle weiteren Klassen in JavaFX

- Jedem Node können weitere Nodes hinzugefügt werden
  - Dies geschieht durch die add-Methode
  - Es entsteht eine **Baumstruktur**
  - Bsp.: Einer Stage wird eine Scene hinzugefügt, einer Scene wird ein Layout hinzugefügt und dem Layout werden Labels, Buttons, TextFields etc. hinzugefügt



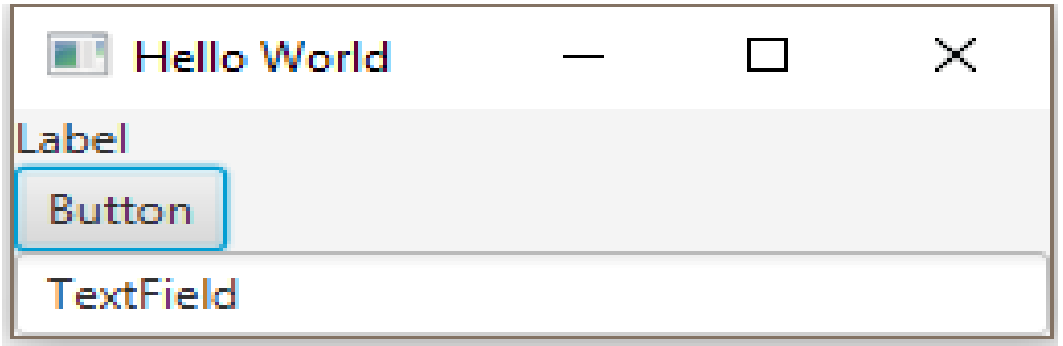
JavaFX - Komponenten Beispiel


```
public void start(Stage primaryStage) throws Exception {
    Label label = new Label("Label");
    Button button = new Button("Button");
    TextField textField = new TextField("TextField");
    VBox layout = new VBox();

    layout.getChildren().addAll(label, button, textField);
    Scene scene = new Scene(layout);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Hello World");

    primaryStage.show();
}
```

Instanziierung des Labels, des Buttons, des TextFields und des Layouts

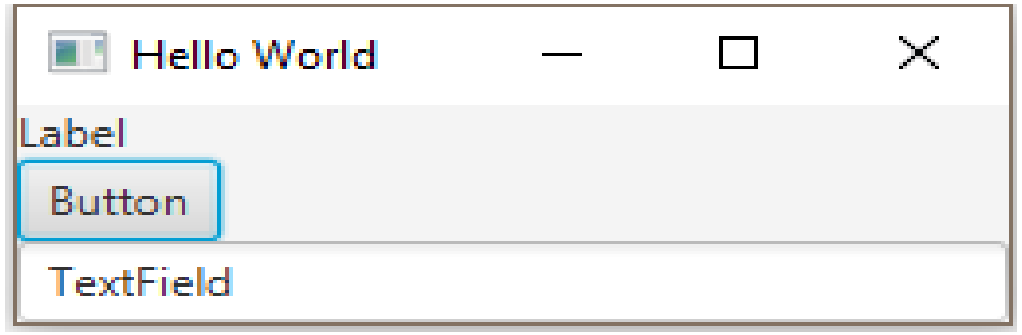
Hinzufügen zum Layout

Instanziierung der Scene, hinzufügen der Scene zur Stage



JavaFX - Komponenten Beispiel


```
public void start(Stage primaryStage) throws Exception {
    Label label = new Label("Label");
    Button button = new Button("Button");
    TextField textField = new TextField("TextField");
    VBox layout = new VBox();

    layout.getChildren().addAll(label, button, textField);
    Scene scene = new Scene(layout);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Hello World");

    primaryStage.show();
}
```



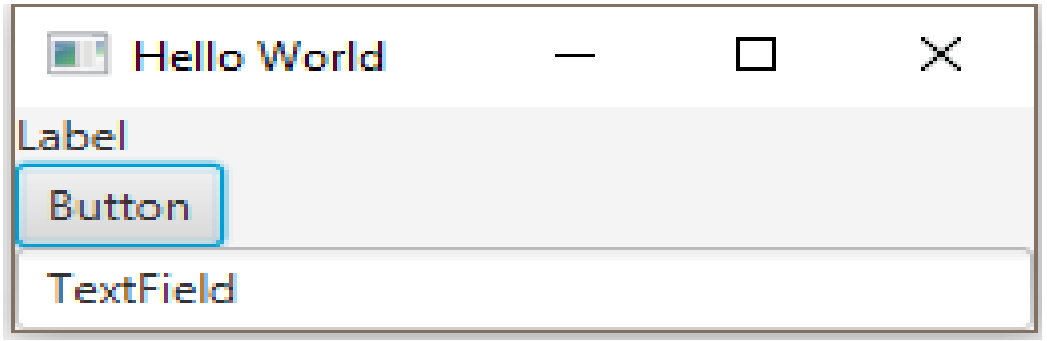
Setzen des Titels der Scene

Anzeigen der kompletten Stage mit all den hinzugefügten Elementen

JavaFX - Komponenten Beispiel


```
public void start(Stage primaryStage) throws Exception {
    Label label = new Label("Label");
    Button button = new Button("Button");
    TextField textField = new TextField("TextField");
    VBox layout = new VBox();

    layout.getChildren().addAll(label, button, textField);
    Scene scene = new Scene(layout);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Hello World");

    primaryStage.show();
}
```

Das Layout bestimmt wie die Komponenten im Fenster angeordnet werden



JavaFX - Layouts

- In JavaFX gibt es eine Vielzahl **verschiedener Layouts**
- Layouts ermöglichen es, Elemente auf einer Szene nach einem vorgefertigten **Schema anzuordnen**
- Gängige Layoutbeispiele sehen wie folgt aus:

TilePane

Comp 1	Comp 2	Comp 3	Comp 4
Comp 5	Comp 6	Comp 7	Comp 8

BorderPane

Comp 1		
Comp 2	Comp 5	Comp 3
Comp 4		

VBox

Comp 1

Comp 2

Comp 3

Comp 4

Agenda

1. Grundlagen
2. Komponenten
3. **Scene Builder**
4. Benutzereingaben/Action Events
5. Anwendung nach dem MVC-Prinzip

JavaFX – Scene Builder

- Mit dem JavaFX Scene Builder können JavaFX GUIs nach dem **WYSIWYG** (“What You See Is What You Get”) Konzept erstellt werden
  - Der Scene Builder kann auf der **Gluon** Homepage heruntergeladen werden
  - Ist ein **eigenständiges Programm**
  - Der Code für GUIs wird **automatisch** generiert
  - Alternativ: FXML manuell schreiben (ähnlich wie HTML)
  - Ist auch in einigen IDEs integrierbar (z.B. IntelliJ IDEA)

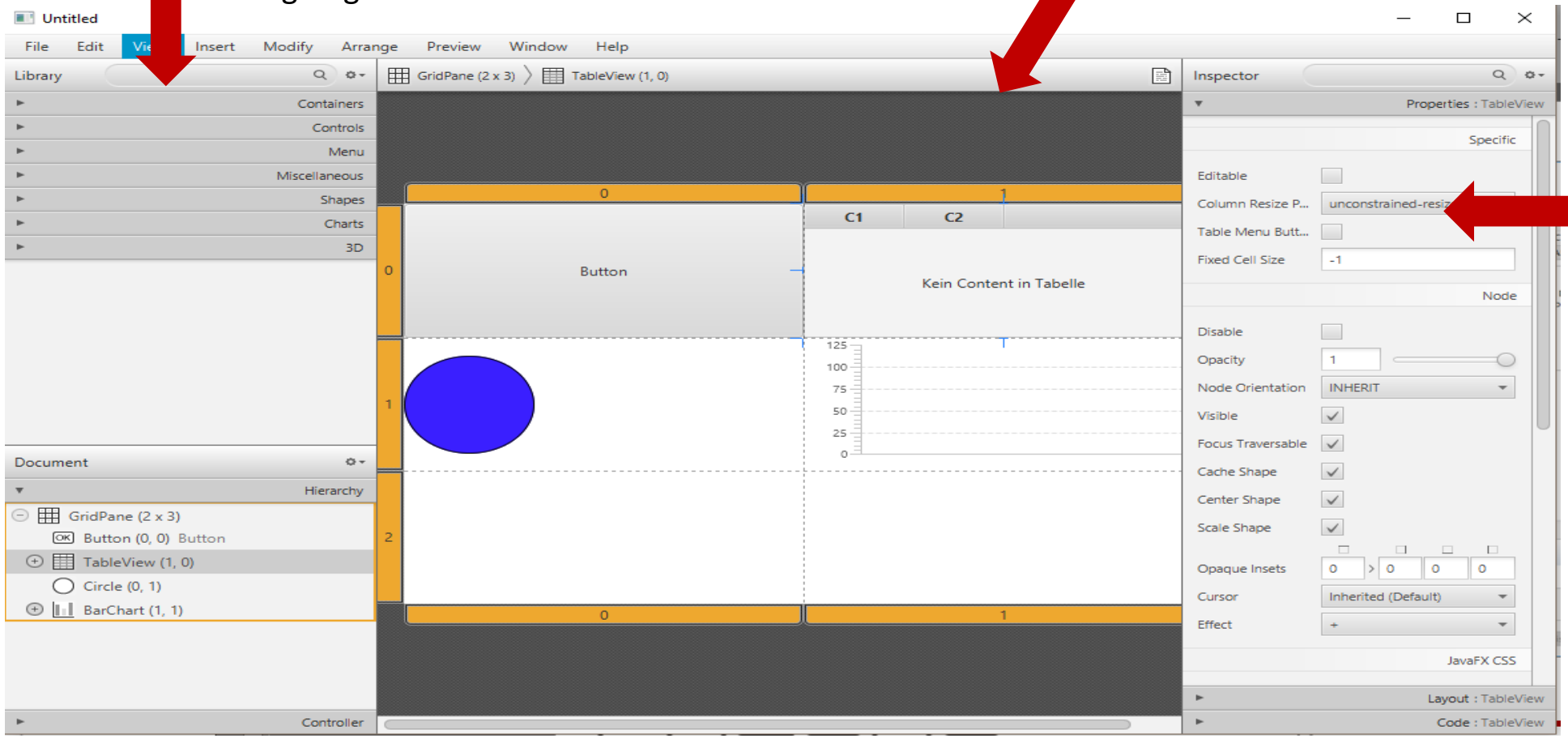
Download unter:

- https://gluonhq.com/products/scene-builder/

JavaFX – Scene Builder Beispiel

Auflistung sämtlicher GUI-Elemente, welche per Drag-And-Drop der GUI hinzugefügt werden können

Vorschau der tatsächlichen GUI



Optionen um ausgewählte Elemente in der GUI anzupassen

JavaFX – Umgang mit Scene Builder/FXML Elementen

- Scene Builder erstellt aus den definierten Elementen eine FXML-Datei, welche den kompletten Baum aller graphischen Elemente, die im Scene Builder hinzugefügt worden sind, darstellt

Festgelegte Position des Elements

Zugehöriger Controller

```
<AnchorPane fx:id="paneLo" prefHeight="600.0" prefWidth="900.0"  xmlns:fx="http://javafx.com/fxml/1" fx:controller="Crawler.LoginController">
   <children>
      <TextField fx:id="usernameField" layoutX="376.0" layoutY="256.0" />
      <PasswordField fx:id="passwordField" layoutX="376.0" layoutY="316.0" />
      <Button fx:id="loginButton" layoutX="466.0" layoutY="375.0" mnemonicParsing="false" onAction="#loginButtonPressed" text="Login" />
      <Text layoutX="242.0" layoutY="113.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Welcome to your S ts-Manger" wrappingWidth="416.003
         <font>
            <Font name="Ch agne &amp;amp; Limousines" size="32.0" />
         </font>
      </Text>
      <Text layoutX="297.0" layoutY="273.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Username" />
      <Text layoutX="299.0" layoutY="333.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Password" />
      <Button fx:id="registerButton" layoutX="386.0" layoutY="375.0" mnemonicParsing="false" onAction="#registerButtonPressed" text="Register"
   </children>
```

Name des Elements

Definiertes Action Event

JavaFX – Umgang mit Scene Builder/FXML Elementen

- Dabei werden die definierten Elemente mittels einer @FXML-Annotation referenziert

FXML-Annotation

```
@FXML
Button loginButton;

@FXML
Button registerButton;

@FXML
TextField usernameField;

@FXML
PasswordField passwordField;
```

Name des im vorherigen Bild gezeigten Elements

Agenda

1. Grundlagen
2. Komponenten
3. Scene Builder
4. **Benutzereingaben/ Action Events**
5. Anwendung nach dem MVC-Prinzip

JavaFX - Behandlung von Benutzereingaben

- **Grundlagen**
  - Alle Benutzereingaben wie z.B. Mausklicks, Mausbewegung und Tastatureingaben werden primär vom Betriebssystem verarbeitet
  - das Betriebssystem leitet diese Eingaben in Form von Nachrichten an das aktuelle Fenster weiter
  - um diese Nachrichten abzufangen, müssen so genannte „Action Events“ implementiert werden
  - ein kurzer Überblick über einige dieser „Action Events“:

Mouse Event	lauscht nach Mausklicks
Action Event	lauscht nach Action Events (z.B. Button gedrückt)
Key Event	lauscht nach Tastatureingaben

JavaFX - Behandlung von Benutzereingaben

- **Beispiel für einen EventHandler:**
  - EventHandler können genutzt werden, um auf das Betätigen von Buttons zu reagieren
  - Wenn mit dem Scene Builder gearbeitet wird, werden diese Events für das jeweilige GUI-Element definiert
  - Die implementierende Klasse muss die im Scene Builder angegebene Methode implementieren
    - z.B.: public void loginButtonPressed(ActionEvent event)
  - Die definierte Methode – in diesem Fall – loginButtonPressed(ActionEvent event) wird dann in der EventQueue ausgeführt, wenn der Button betätigt wird

JavaFX - Behandlung von Benutzereingaben - Beispiel

Methodenname für das Action Event

Weitere Auswahlmöglichkeiten für Benutzereingaben

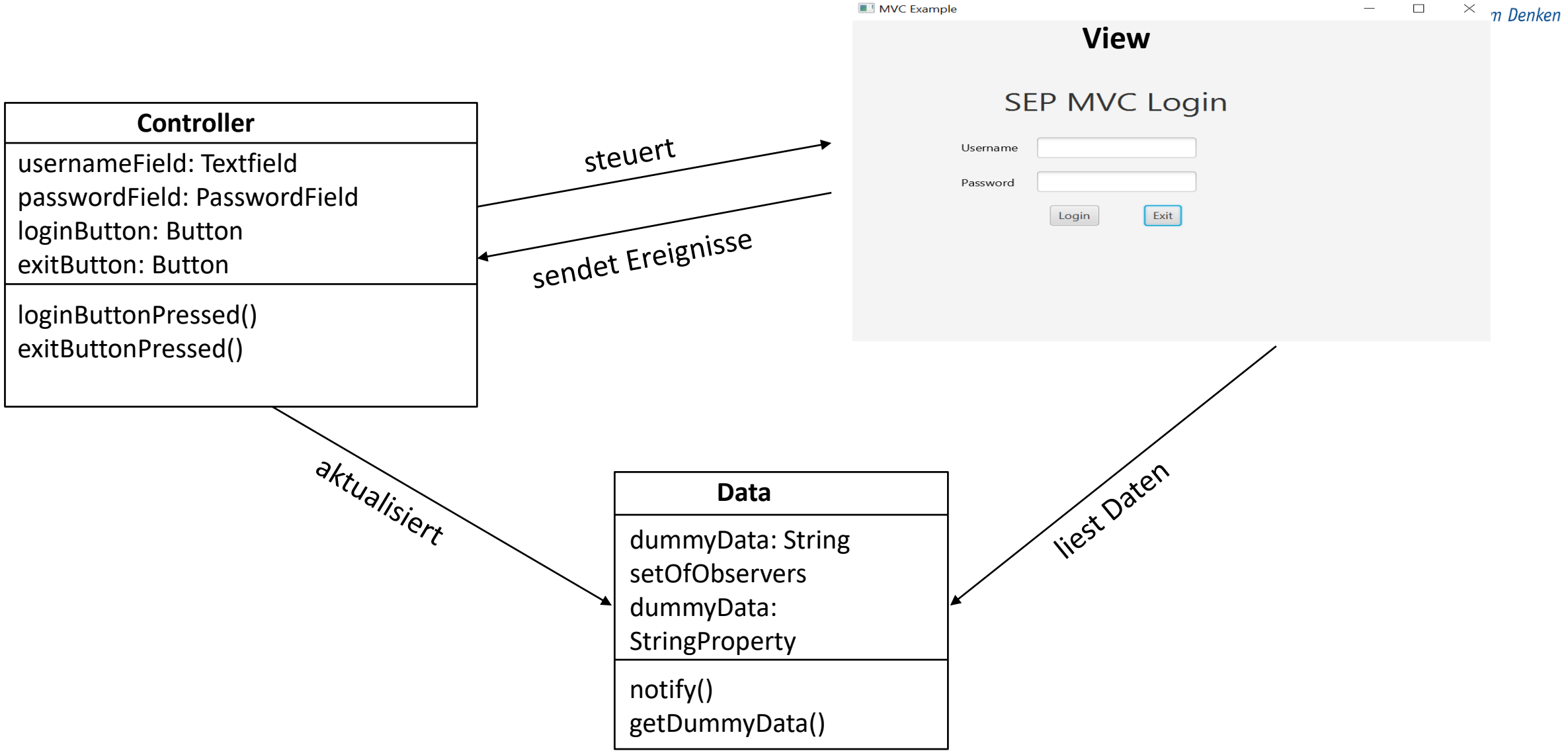
Name des Action Events (festgelegt im Scene Builder)

```
public void loginButtonPressed() {
    usernameField.clear();
}
```

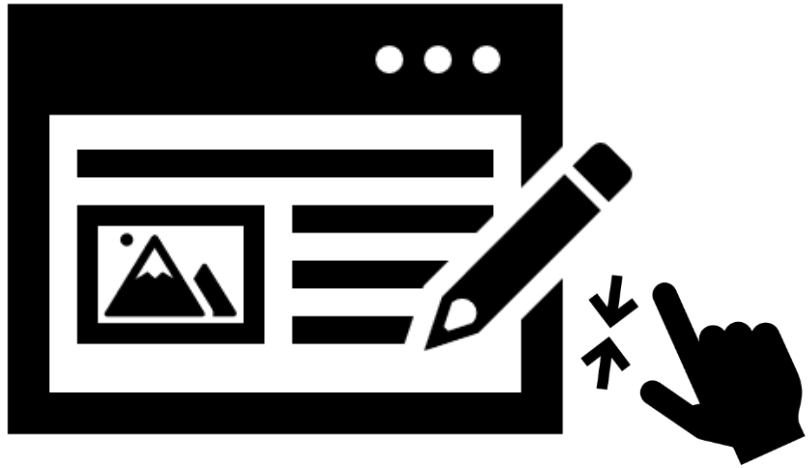
Agenda

1. Grundlagen
2. Komponenten
3. Scene Builder
4. Benutzereingaben/Action Events
5. **Anwendung nach dem MVC-Prinzip**

Ausblick: Arbeiten mit JavaFX nach dem MVC-Prinzip



Hands-On JavaFX



JavaFX – Und jetzt?

- https://openjfx.io/
  - Vollständige Dokumentation
  - Unter dem Reiter „**Community**“ findet man viele nützliche **tools and frameworks**
  - https://openjfx.io/openjfx-docs/
    - Zeigt einen vollständigen Starting Guide für das Arbeiten mit OpenJFX

- Empfehlung: **BootstrapFX** (Design der Benutzeroberfläche)

Quellen

- https://commons.wikimedia.org/wiki/File:Javafx-stage-scene-node.jpg
- https://jaxenter.com/netbeans/developing-nasas-mission-software-with-java
- https://dzone.com/refcardz/javafx-8-1
- http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-1x-archive-2199384.html#javafx-scenebuilder-2.0-oth-JPR
- https://docs.oracle.com/javafx/2/events/processing.htm#CEGJAAFD
- https://docs.oracle.com/javase/8/javafx/api/overview-tree.html
- https://openjfx.io/
- https://javabeginners.de/Frameworks/JavaFX/FXML.php
- http://www.datenbanken-verstehen.de/lexikon/model-view-controller-pattern/

UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

Vielen Dank für Ihre Aufmerksamkeit

PALUNO
The Ruhr Institute for Software Technology