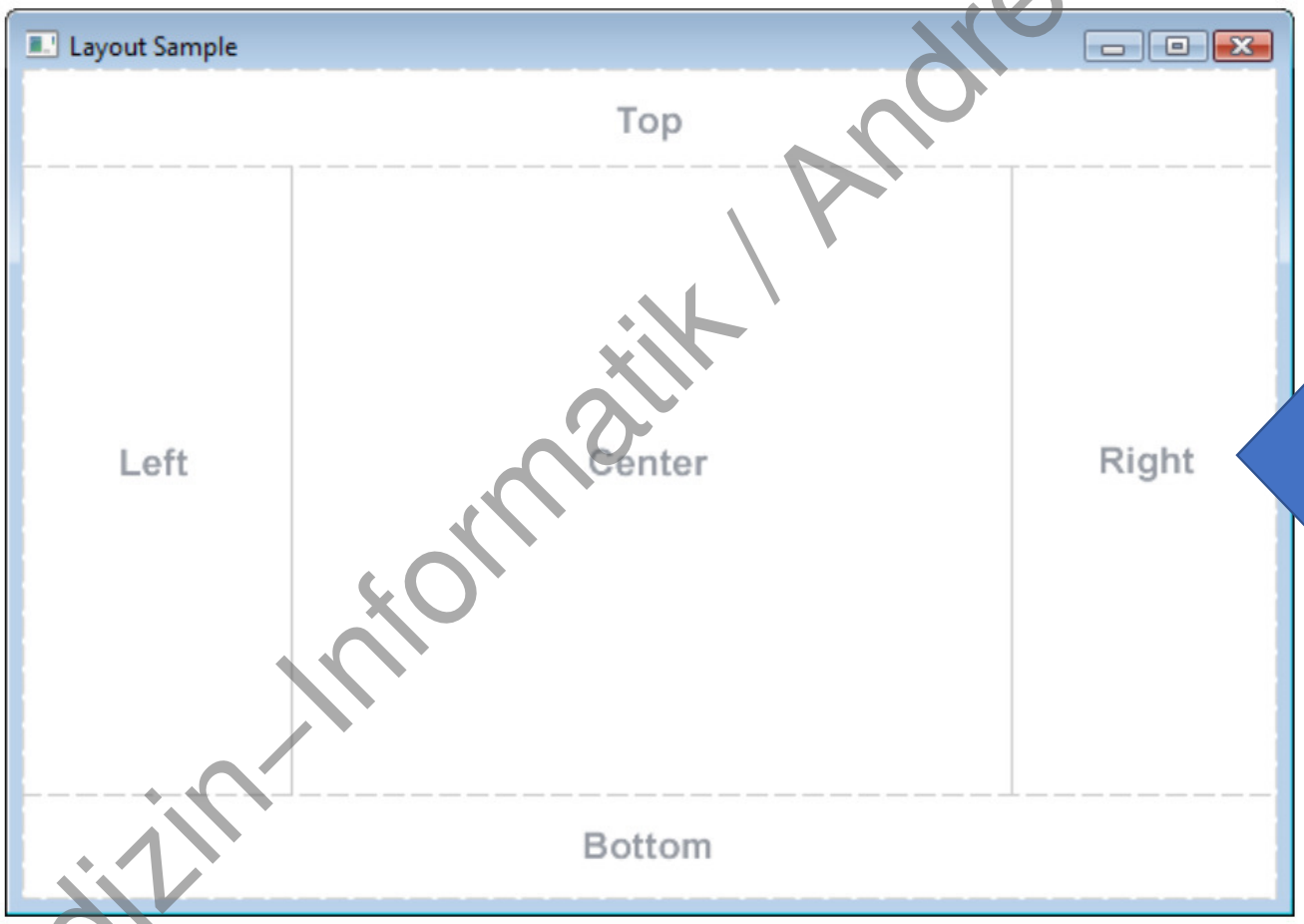
Java
Container

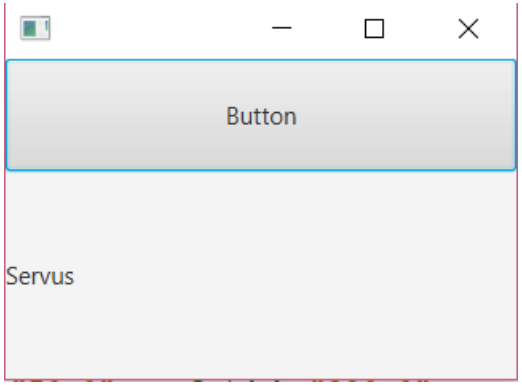
Eine Oberfläche mit Container strukturieren

- Das Layout einer Java-FX Anwendung sollte immer mit Container gestaltet werden.
- Der Sinn einer vernünftigen Gestaltung liegt darin, dass eine Veränderung der Größe des Fensters die Komponenten trotzdem relativ zum Fenster platziert.
- Man spricht hier gerne von ‚Responsive Design‘

BorderPane



Das Layout bietet 5 Regionen an. Die Regionen können eine beliebige Größe haben. Wird eine Region nicht benötigt, so wird diese einfach nicht definiert.



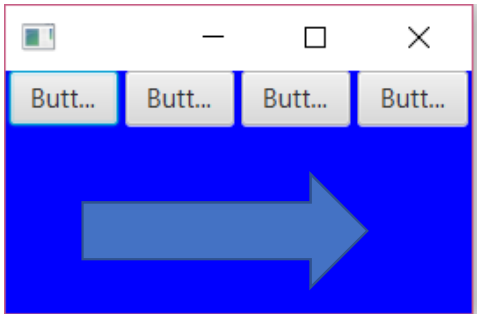
BorderPane

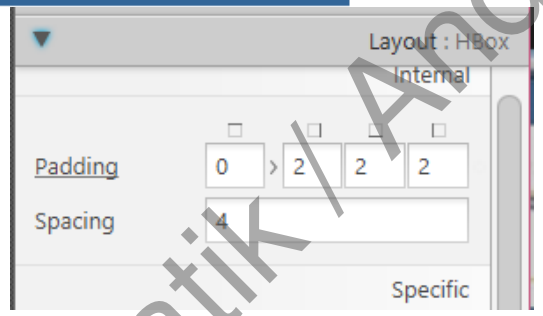
```
<BorderPane id="BorderPane" prefHeight="200" prefWidth="320"
            xmlns="http://javafx.com/javafx/8.0.111" xmlns:fx="http://javafx.com/fxml/1"
            fx:controller="vorzeigen.FXMLDocumentController">
   <top>
      <Button mnemonicParsing="false" prefHeight="70.0" prefWidth="320.0" text="Button"
              BorderPane.alignment="CENTER" />
   </top>
   <left>
      <Label prefHeight="88.0" prefWidth="84.0" text="Servus" BorderPane.alignment="CENTER" />
   </left>
</BorderPane>
```

Das Layout bietet 5 Regionen an. Die Regionen sind durch top, left, right, bottom und center definiert.

Projekt -> Vorzeigen (FXMLBorderPane)

HBox



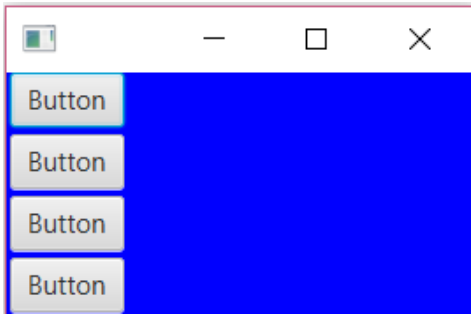


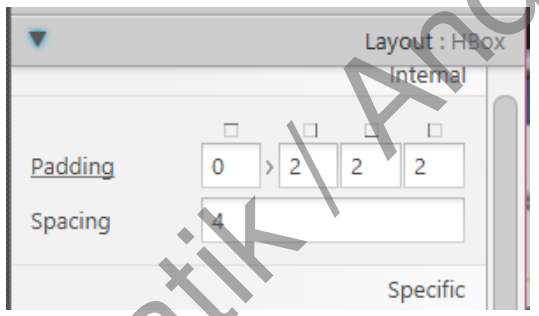
Mit HBox kann man Elemente in einer Reihe anordnen (Abstand zwischen den Elementen kann definiert werden)

```
<HBox id="HBox" prefHeight="138.0" prefWidth="265.0" spacing="4.(
   <children>
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
   </children>
   <padding>
      <Insets bottom="2.0" left="2.0" right="2.0" />
   </padding>
</HBox>
```

Projekt -> Vorzeigen (FXMLHBox.fxml)

VBox



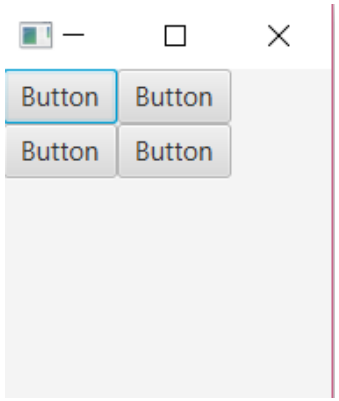


Mit VBox kann man Elemente in einer Reihe anordnen (Abstand zwischen den Elementen kann definiert werden)

```
<VBox id="VBox" prefHeight="138.0" prefWidth="265.0" spacing="4.0" style="-fx-background-color: blue;"
   <children>
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
   </children>
   <padding>
      <Insets bottom="2.0" left="2.0" right="2.0" />
   </padding>
</VBox>
```

Projekt -> Vorzeigen (FXMLVBox.fxml)

FlowPane

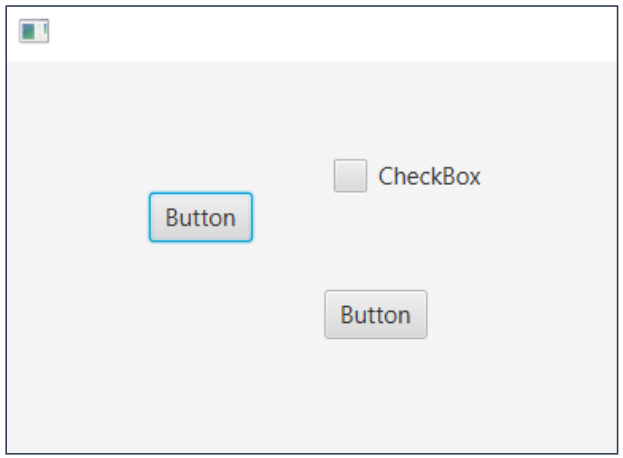


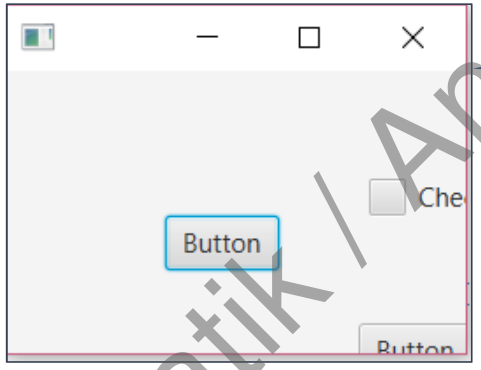
Mit FlowPane werden der Reihe nach die Elemente aufgelistet. Ist kein Platz mehr für ein Element, so wird eine neue Zeile begonnen.

```
<FlowPane id="FlowPane" prefHeight="400.0" prefWidth="187.0" xmlns
    <children>
            <Button mnemonicParsing="false" text="Button" />
            <Button mnemonicParsing="false" text="Button" />
            <Button mnemonicParsing="false" text="Button" />
            <Button mnemonicParsing="false" text="Button" />
        </children>

</FlowPane>
```

Projekt -> Vorzeigen (FXMLVBox.fxml)

AnchorPane



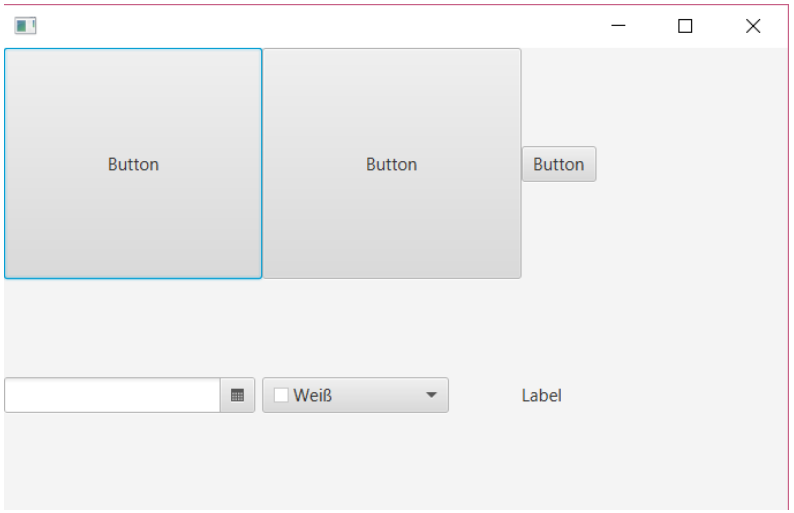


Mit AnchorPane werden Elemente absolut zum obersten Eckpunkt notiert. Wird das Fenster verkleinert, so werden die Punkte nicht in der Position verändert.

```
<AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns:fx="http://javafx.com/
   <children>
      <Button layoutX="91.0" layoutY="82.0" mnemonicParsing="false" text="Button" />
      <Button layoutX="201.0" layoutY="143.0" mnemonicParsing="false" text="Button" />
      <CheckBox layoutX="207.0" layoutY="61.0" mnemonicParsing="false" text="CheckBox" />
   </children>
</AnchorPane>
```

Projekt -> Vorzeigen (FXMLAnchorPane.fxml)

GridPane

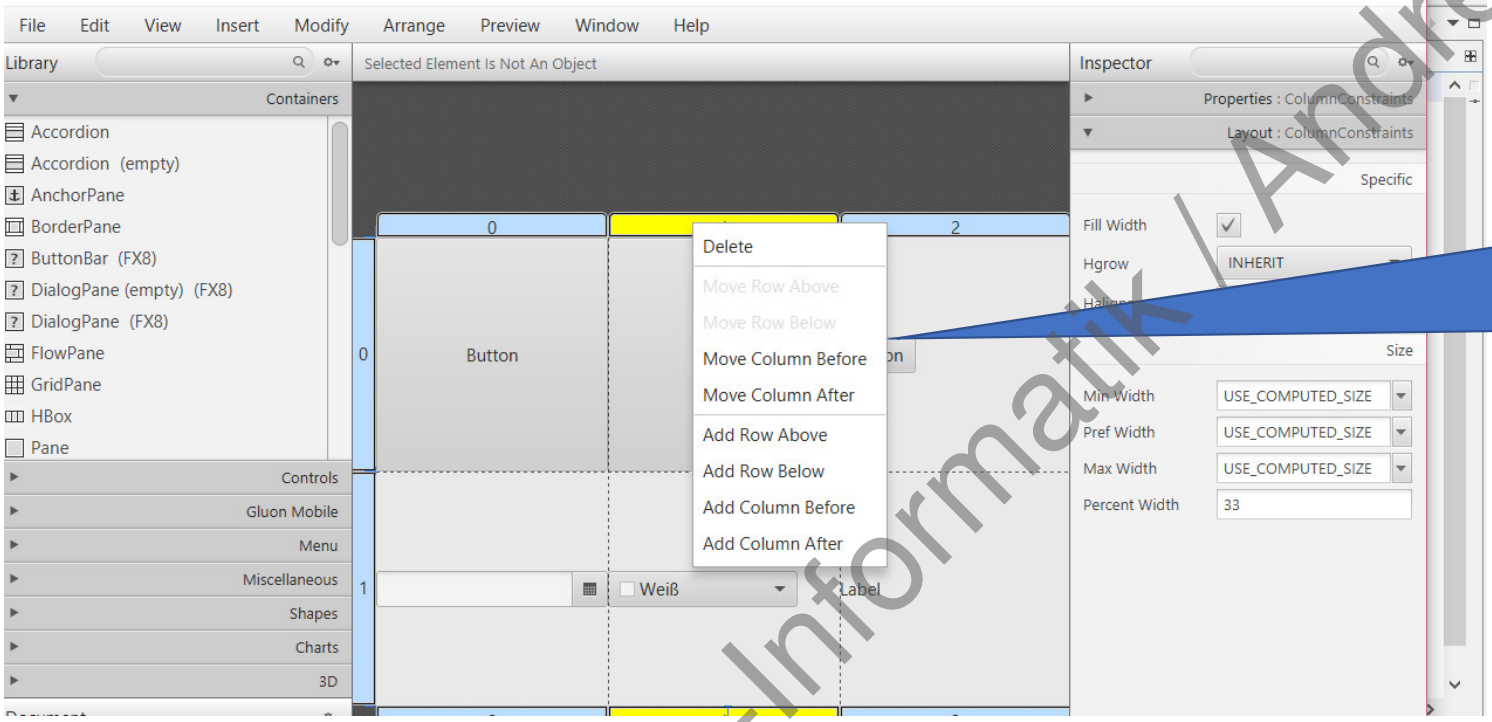



```
<GridPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/
   <columnConstraints>
      <ColumnConstraints percentWidth="33.0" />
      <ColumnConstraints percentWidth="33.0" />
      <ColumnConstraints percentWidth="33.0" />
   </columnConstraints>
   <rowConstraints>
      <RowConstraints percentHeight="50.0" />
      <RowConstraints minHeight="10.0" percentHeight="50.0" prefHeight="30.0" />
   </rowConstraints>
   <children>
      <Button maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" mnemo
      <Button maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" mnemo
      <Button mnemonicParsing="false" text="Button" GridPane.columnIndex="2" />
      <ColorPicker GridPane.columnIndex="1" GridPane.rowIndex="1" />
      <DatePicker GridPane.rowIndex="1" />
      <Label text="Label" GridPane.columnIndex="2" GridPane.rowIndex="1" />
   </children>
</GridPane>
```

Mit GridPane können Spalten und Zeilen definiert werden und diese mit Elementen (vielleicht auch Panes) gefüllt werden

Projekt -> Vorzeigen (FXMLGridPane.fxml)

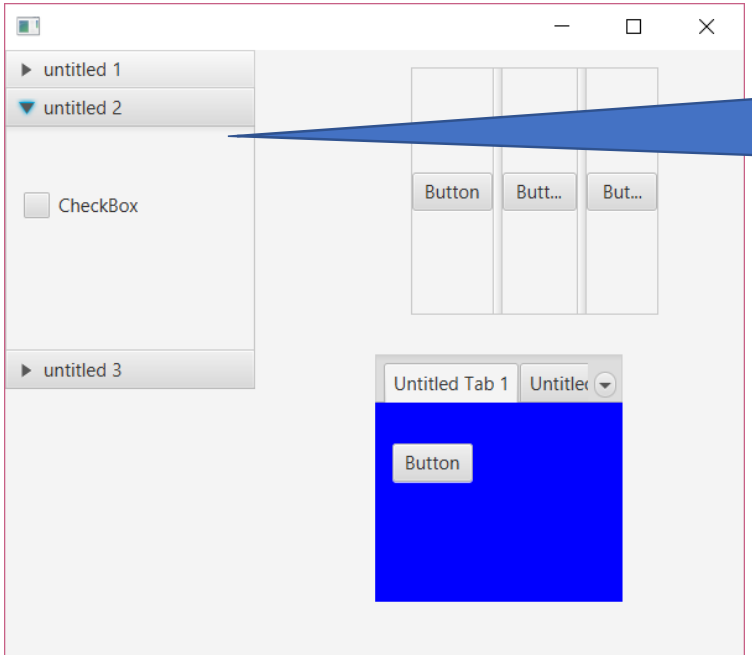
GridPane



Im Scene Builder können Spalten entfernt und hinzugefügt werden (bzw. Reihen)

Projekt -> Vorzeigen (FXMLGridPane.fxml)

Weitere Panes

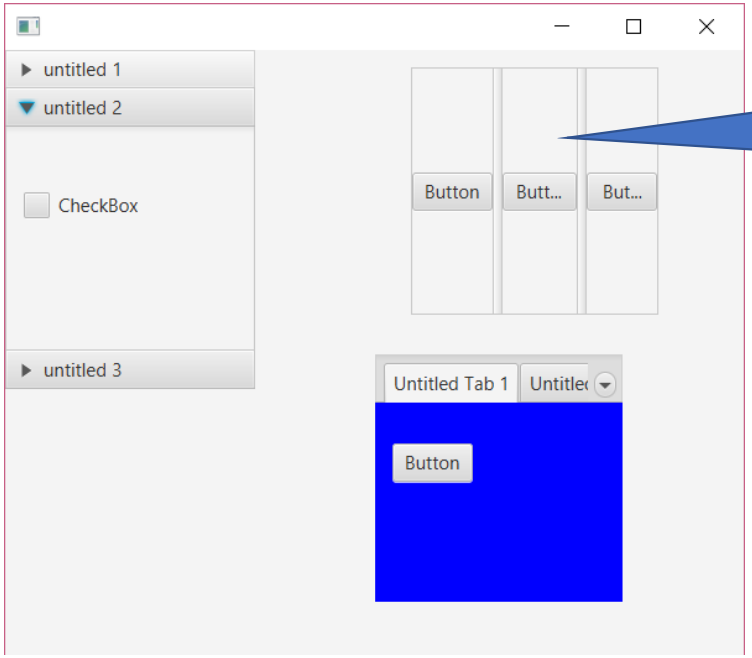


Das Accordion ermöglicht das Aufklappen von Elementen. Jede Ebene ist durch eine TitledPane beschrieben.

```
<Accordion>
  <panes>
    <TitledPane animated="false" text="untitled 1">
      <content>
        <AnchorPane>
      </content>
    </TitledPane>
    <TitledPane animated="false" text="untitled 2">
      <content>
        <AnchorPane>
      </content>
    </TitledPane>
    <TitledPane animated="false" text="untitled 3">
      <content>
    </TitledPane>
  </panes>
</Accordion>
```

Projekt -> Vorzeigen (FXMLPanes.fxml)

Weitere Panes

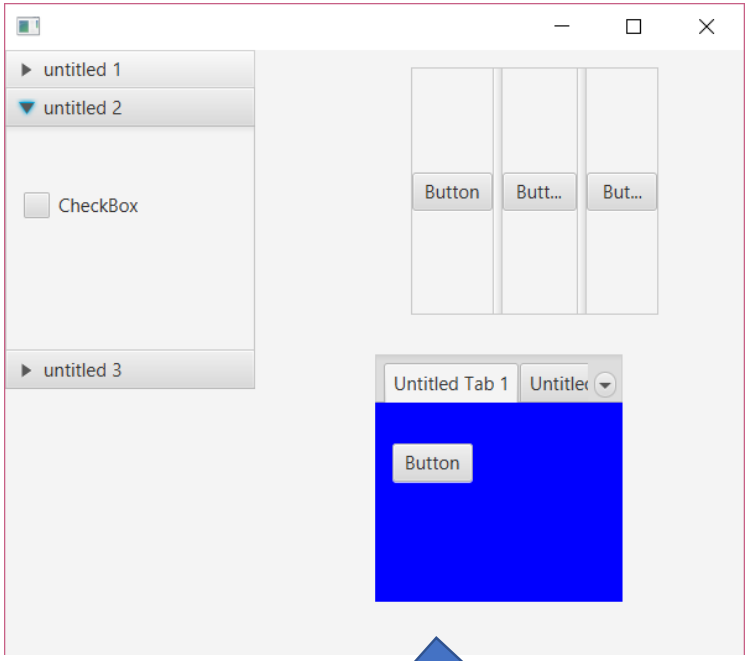


Die SplitPane ermöglicht das Verschieben einer Trennlinie zwischen verschiedenen Arbeitsblätter

```
<SplitPane dividerPositions="0.3484848484848485, 0.6919191919191919" layoutX="3
   <items>
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
      <Button mnemonicParsing="false" text="Button" />
   </items>
</SplitPane>
```

Projekt -> Vorzeigen (FXMLPanes.fxml)

Weitere Panes

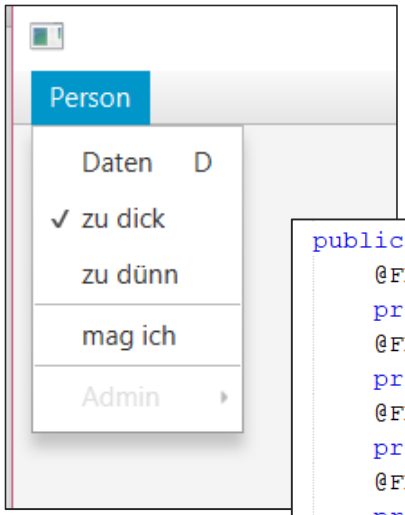



```
<TabPane layoutX="299.0" layoutY="246.0" prefHeight="200.0" prefWidth="200.0" tabClosing
  <tabs>
    <Tab text="Untitled Tab 1">
      <content>
        <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0"
            <children>
              <Button layoutX="14.0" layoutY="33.0" mnemonicParsing="false" text="Bu
            </children>
          </AnchorPane>
      </content>
    </Tab>
    <Tab text="Untitled Tab 2">
      <content>
        <AnchorPane>
      </content>
    </Tab>
  </tabs>
</TabPane>
```

Die TabPane ermöglicht das Umschalten zwischen Reitern.

Projekt -> Vorzeigen (FXMLPanes.fxml)

Menüs

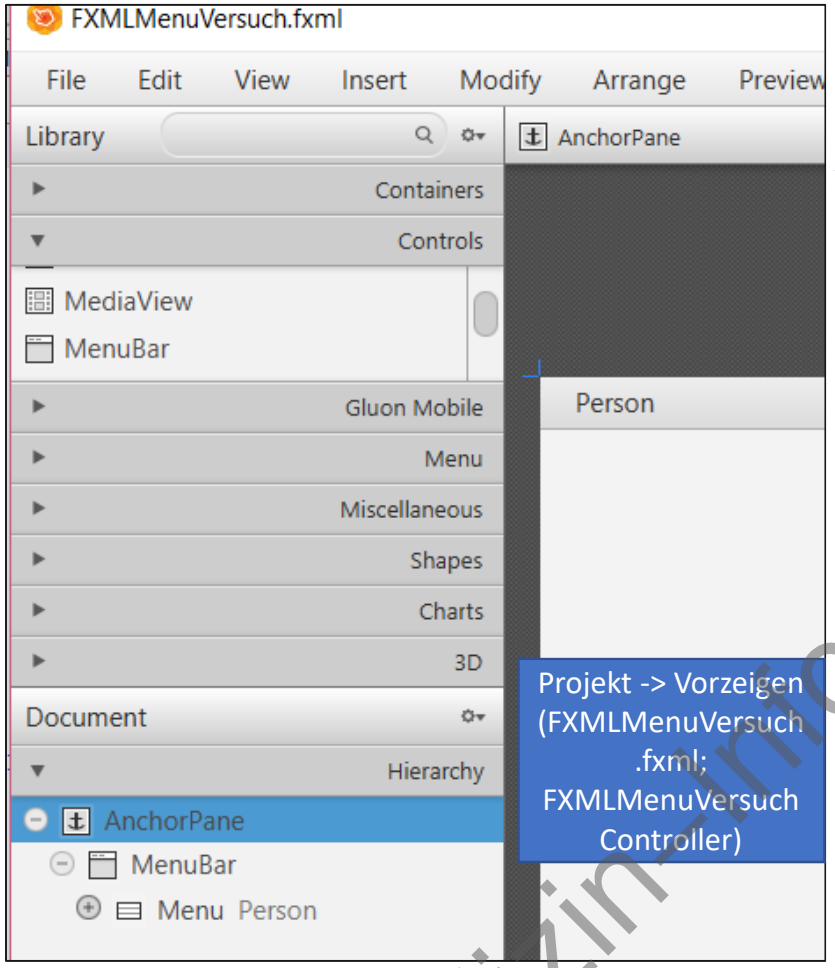



```
public class FXMLMenuVersuchController implements Initializable {
    @FXML
    private ToggleGroup testGroup;
    @FXML
    private RadioMenuItem menuItem_zuDick;
    @FXML
    private RadioMenuItem menuItem_ZuDünn;
    @FXML
    private CheckMenuItem checkMenuItem_MagIch;
    @FXML
    private Menu menu_Admin;

    /** Initializes the controller class ...3 lines */
    @Override
    public void initialize(URL url, ResourceBundle rb) {...3 lines }

    @FXML
    private void datenAnzeigen(ActionEvent event) {
        System.out.println("Daten wurde betätigt");
        System.out.println("Zu_Dick?" + menuItem_zuDick.isSelected());
        System.out.println("zu_Dünn?" + menuItem_ZuDünn.isSelected());
        checkMenuItem_MagIch.setSelected(true);
        if (menuItem_ZuDünn == testGroup.getSelectedToggle()) {
            System.out.println("Abfrage über Toggle auf zuDünn =" + true);
        } else if (menuItem_zuDick == testGroup.getSelectedToggle()) {
            System.out.println("Abrage über Toggle auf zuDick =" + true);
        }
        menu_Admin.setDisable(false);

    }
```

Wir betrachten das hier dargestellte Menü mit MenuItems, RadioMenuItems und CheckMenuItems.

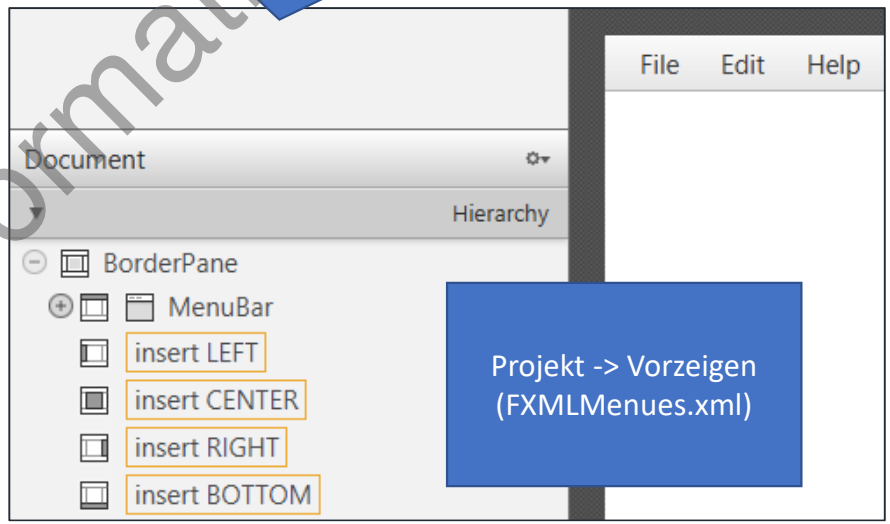
Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

Menüs



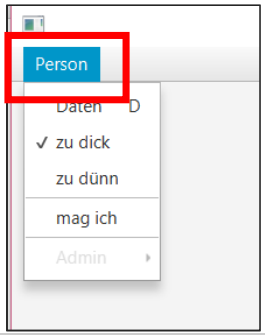
Wir brauchen für das Positionieren der Menues einen MenuBar. Dieser sollte in einem entsprechenden Container platziert werden, sodass dieser sich mit der Größe des Fensters ändert

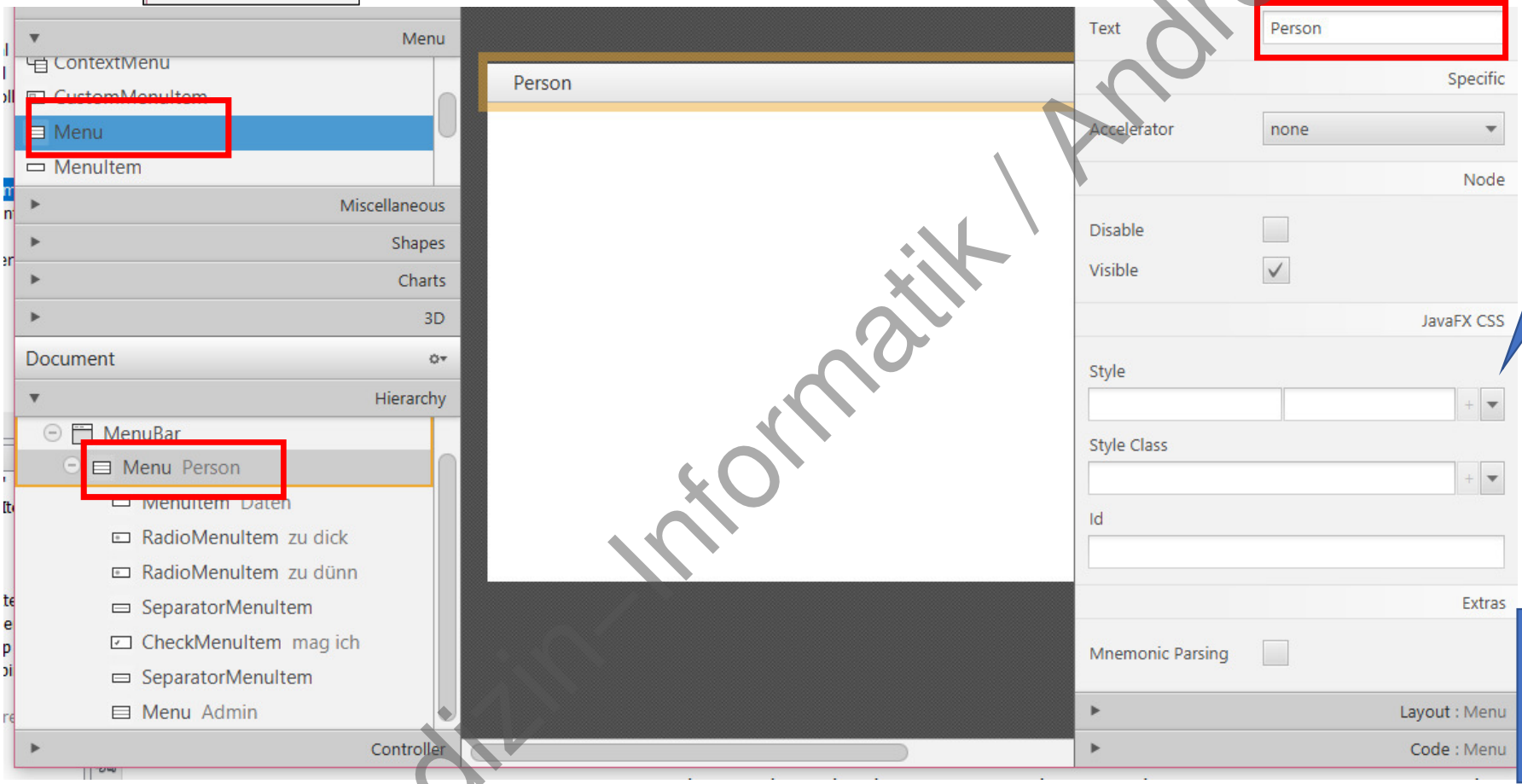
Projekt -> Vorzeigen (FXMLMenuVersuch .fxml; FXMLMenuVersuch Controller)



Projekt -> Vorzeigen (FXMLMenues.xml)

Menüs

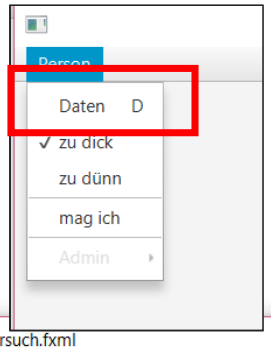


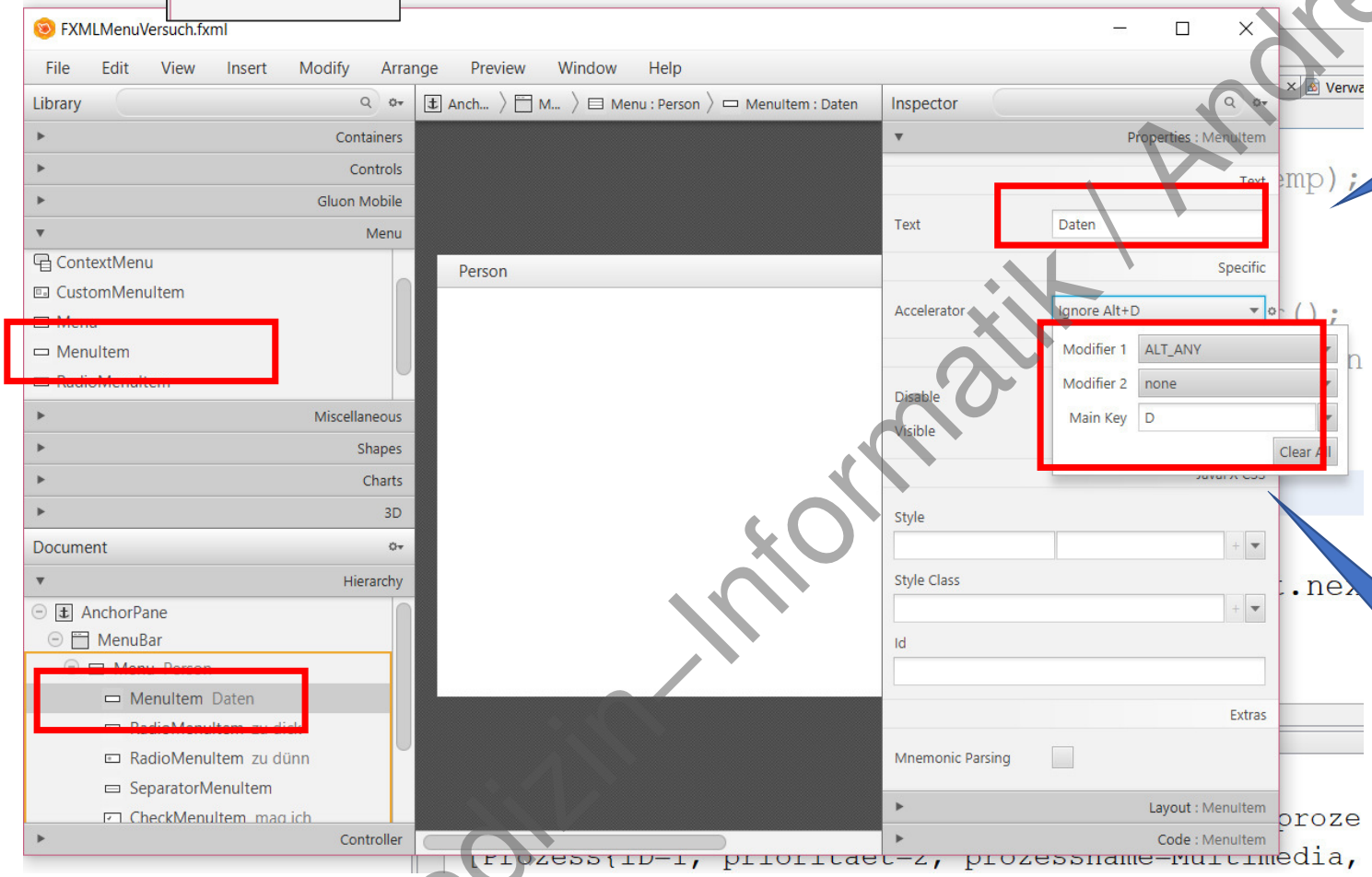


Ein Menubar enthält Menus. Diese können aufgeklappt werden und weitere Komponenten, auch wieder Menus enthalten.

Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

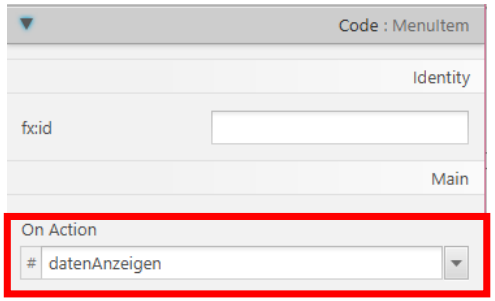
Menüs





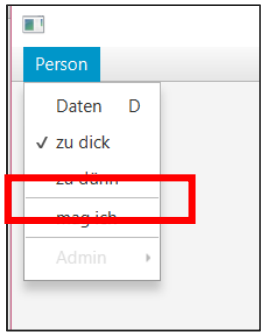
Ein MenuItem ist ein Menüeintrag, welcher keine Unterpunkte mehr hat. Es kann ein Shortcut erstellt werden

Click-Event



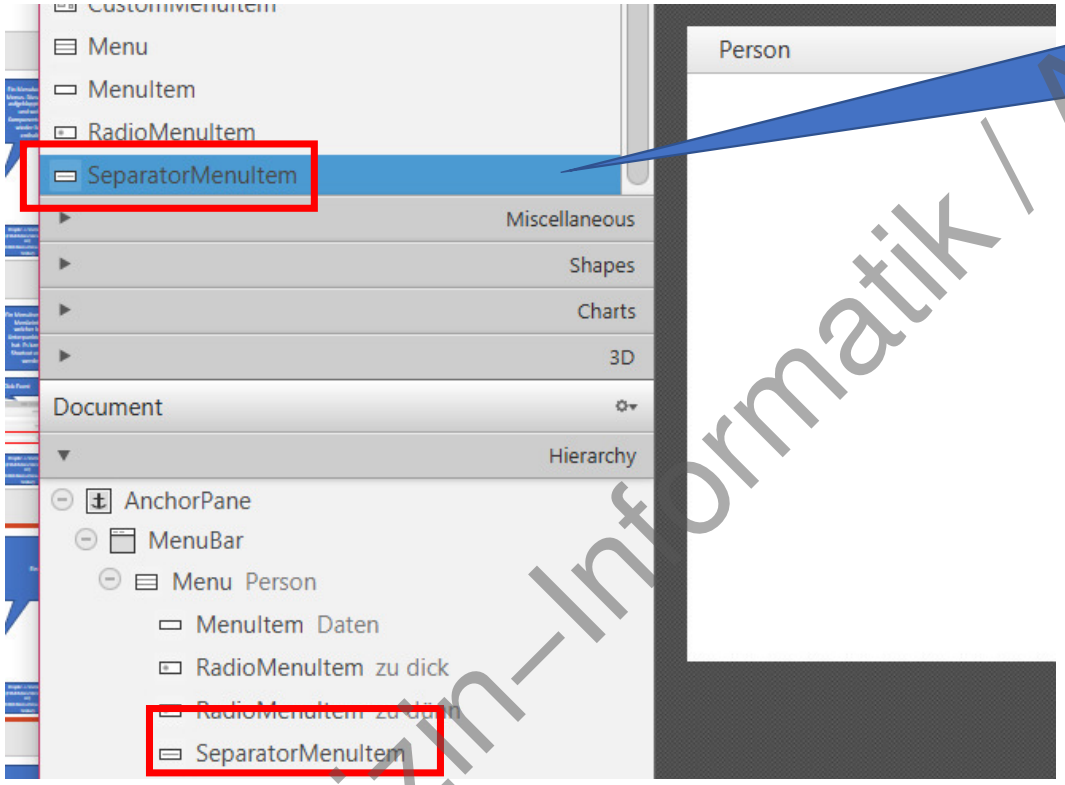
Shortcut

Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)



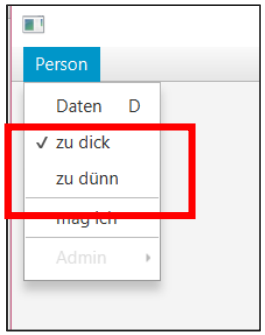
Menüs

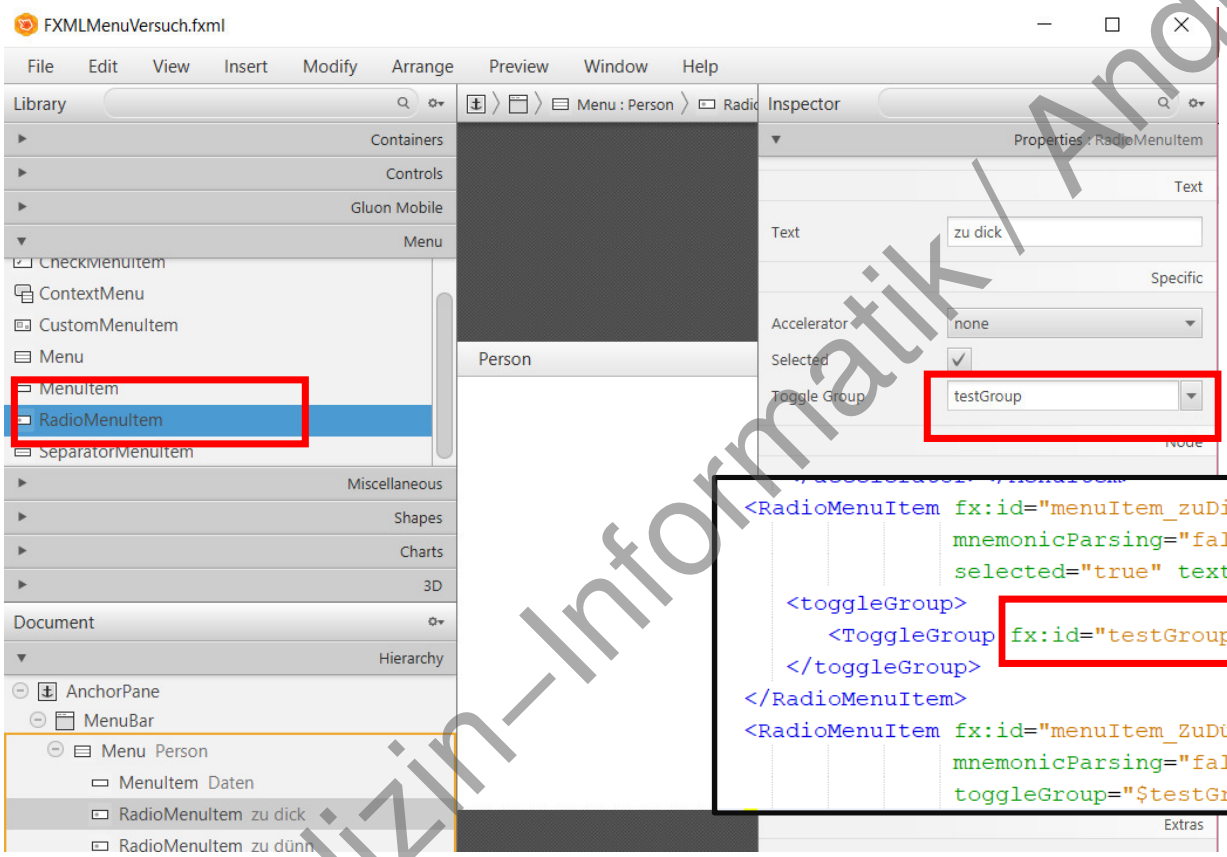
SeparatorMenuItem ist eine Trennlinie



Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

Menüs

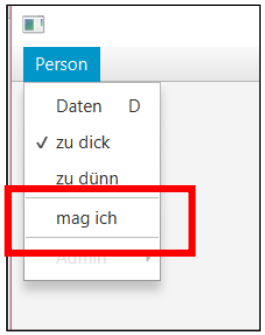




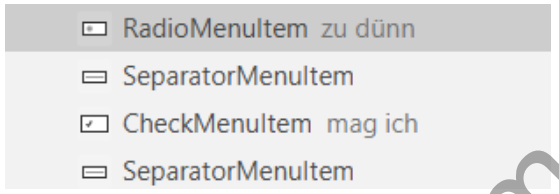

```
<RadioMenuItem fx:id="menuItem_zuDick"
               mnemonicParsing="false"
               selected="true" text="zu dick">
   <toggleGroup>
      <ToggleGroup fx:id="testGroup" />
   </toggleGroup>
</RadioMenuItem>
<RadioMenuItem fx:id="menuItem_ZuDünn"
               mnemonicParsing="false" text="zu dünn"
               toggleGroup="$testGroup" />
```

Das RadioMenuitem kann selektiert und deselektiert werden, mehrere RadioMenuItems können von einer Toogle-Group verwaltet werden, diese verhindert, dass mehrere selektiert sind (Toogle-Group einfach freien Namen wählen – Objekt wird automatisch erstellt

Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)



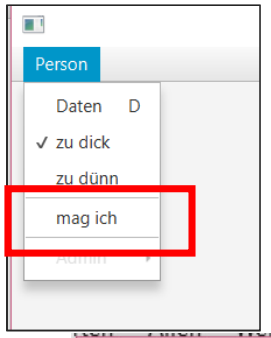
Menüs



CheckMenuItem ist von der Funktion wie das RadioMenuItem

Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

Menüs



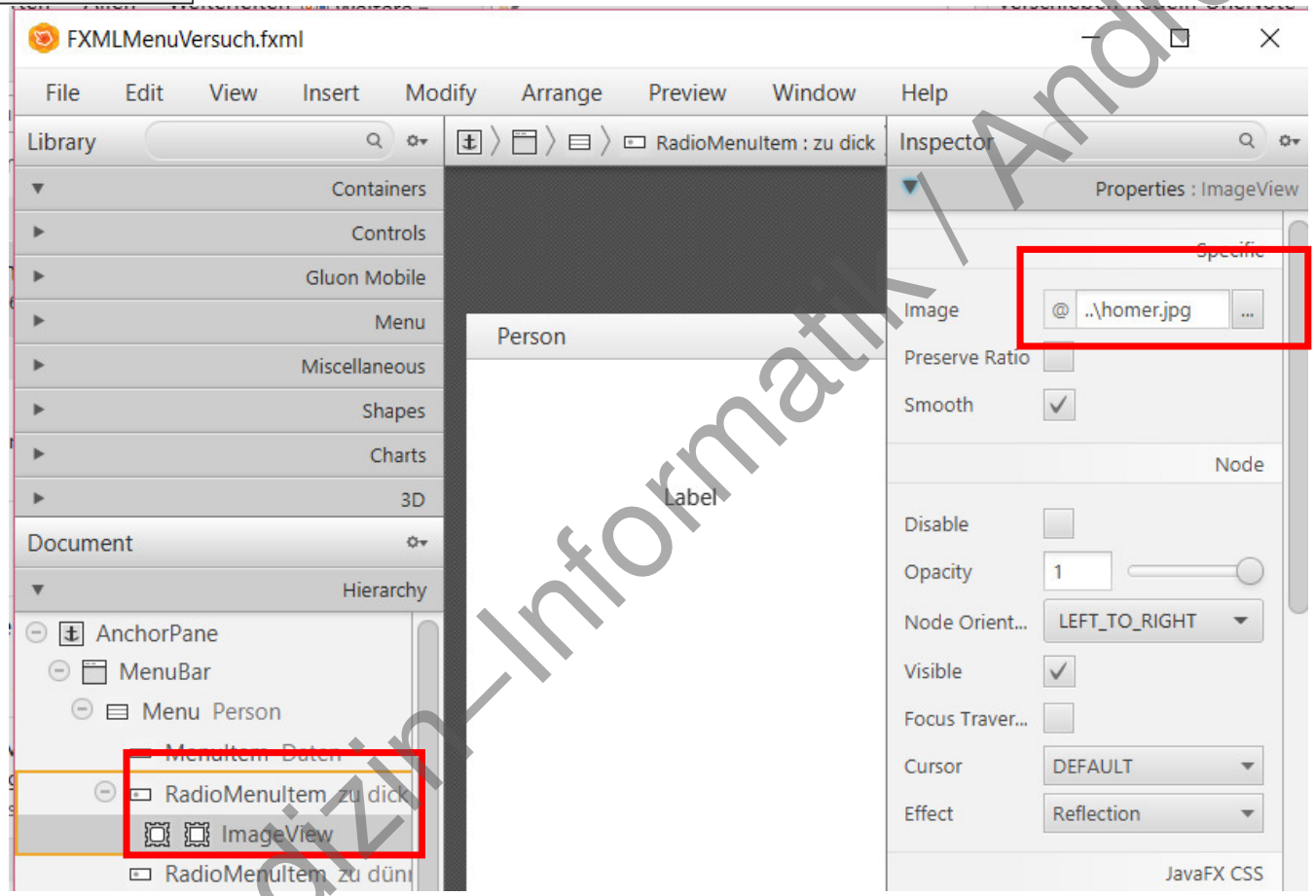
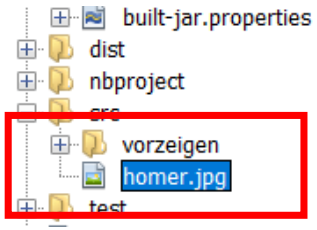


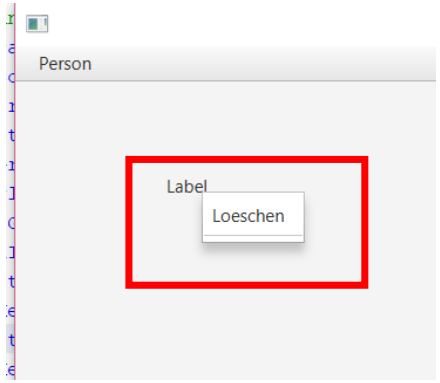
Bild einfügen bei Menü

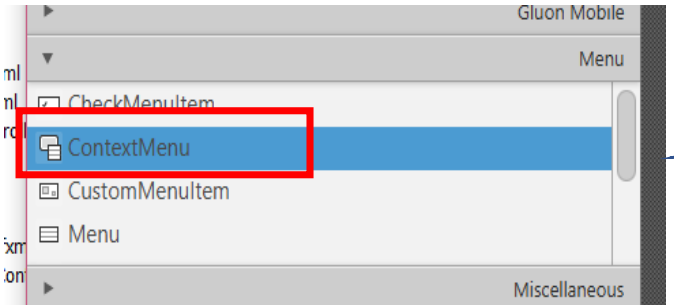
Man zieht eine ImageView auf das Menü und referenziert das Bild. Dies muss im source-Ordner liegen (oder Unterordner)



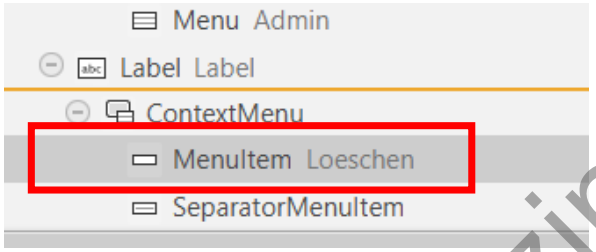
Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

Kontext-Menü



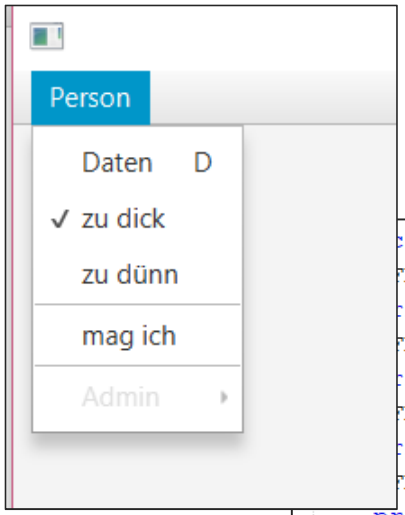


Das Kontextmenü wird auf ein Element gezogen und ist diesem somit zugeordnet. Wird die rechte Maustaste betätigt, so wird diese geöffnet.
Es können die bereits besprochenen Komponenten der Menüs verwendet werden.



Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

Menüs




```
class FXMLMenuVersuchController implements Initializable {
    @FXML
    private ToggleGroup testGroup;
    @FXML
    private RadioMenuItem menuItem_zuDick;
    @FXML
    private RadioMenuItem menuItem_ZuDünn;
    @FXML
    private CheckMenuItem checkMenuItem_MagIch;
    @FXML
    private Menu menu_Admin;

    /** Initializes the controller class ...3 lines */
    @Override
    public void initialize(URL url, ResourceBundle rb) {...3 li }

    @FXML
    private void datenAnzeigen(ActionEvent event) {
        System.out.println("Daten wurde betätigt");
        System.out.println("Zu_Dick?" + menuItem_zuDick.isSelected());
        System.out.println("zu_Dünn?" + menuItem_ZuDünn.isSelected());
        checkMenuItem_MagIch.setSelected(true);
        if (menuItem_ZuDünn == testGroup.getSelectedToggle()) {
            System.out.println("Abfrage über Toggle auf zuDünn =" + true);
        } else if (menuItem_zuDick == testGroup.getSelectedToggle()) {
            System.out.println("Abrage über Toggle auf zuDick =" + true);
        }
        menu_Admin.setDisable(false);

    }
```

MenuRadioItem, CheckMenuItem kann somit überprüft werden, ob es selektiert ist.

getSelectedToggle liefert das MenuItem, welches selektiert sit.

Menü ist nicht mehr selektierbar

Projekt -> Vorzeigen (FXMLMenuVersuch.fxml; FXMLMenuVersuchController)

Are there any Questions?

Exercise:

B_Uebung

C_Uebung