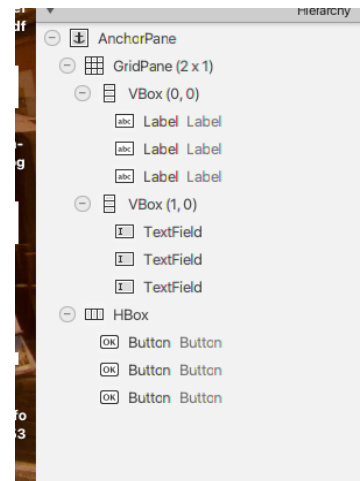
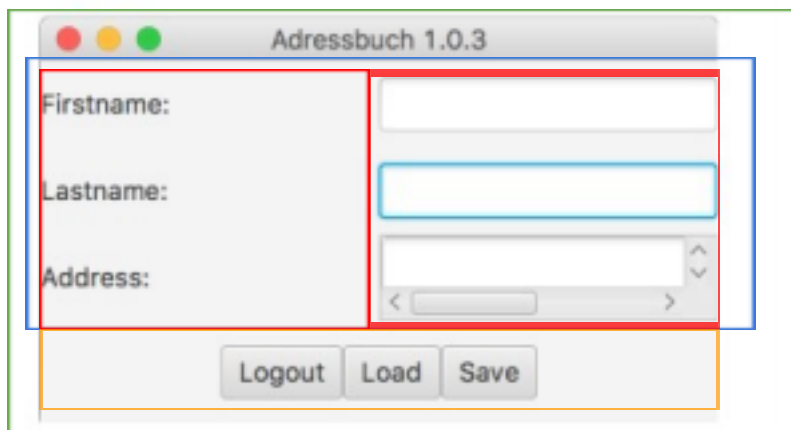


Nachdenkzettel zu GUI Programmierung

Sara Tietze (40800), Pia Schilling(40905), Merve Özdemir (41709)

1. Welche Layout-Manager würden Sie verwenden, um das folgende Fenster zu realisieren (Abbildung 1)? Zeichnen Sie die Layout-Manager direkt in die Abbildung. Tipp: Folgende Layout-Manager wurden in der Vorlesung besprochen: `BorderPane`, `HBox`, `VBox`, `StackPane`, `GridPane`, `FlowPane`, `TilePane`.



`Hbox` `VBox` `GridPane` `AnchorPane`



2. Eventhandler

a) Geben Sie ein Beispiel für die zwei Formen von JavaFX Eventhandlern (alt und seit Java8)

Filter:

```
Button button2 = new Button("2");
button2.addTaskFilter(ActionTask.ACTION,
new TaskHandler<Task>() {
    @Override
    public void handle(Task task) {
        task.consume();
    }
});
```

```
button2.setOnAction( task -> button2.setText("Clicked"));
button2.setOnAction( task -> button2.setText("Clicked"));
```

Handler:

```
Button button1 = new Button("Hey");
button1.setOnAction(new TaskHandler<ActionTask>() {
    @Override
    public void handle(ActionTask task) {
        button1.setText("Clicked");
    }
});
```

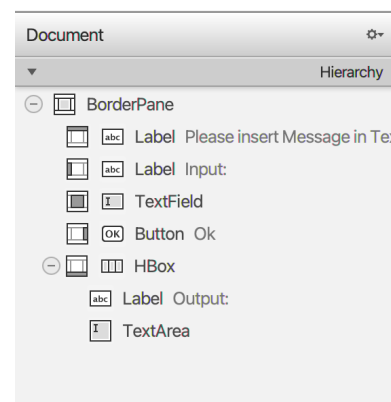
b) Welche Möglichkeit gibt es in JavaFX, nach einer bestimmten Zeit einen Handler-Callback zu bekommen? Wozu könnte man das verwenden?

Datenbankabfragen im Hintergrund laufen lassen und den GUI Thread darüber "informieren" sobald die Daten geladen sind. So kann der GUI Thread weiterarbeiten während die Daten geladen werden.

Dafür gibt es die Methode `Platform.runLater`

3. GUI:

1. Zeichnen Sie das Fenster mit Inhalt, das durch den JavaFX Code beschrieben wird.



2. Was macht der Handler?

Gibt den Input aus dem Textfeld inputArea, in dem Outputfeld outputArea nacheinander aus und schreibt "You:" davor

```
public class LayoutExample extends Application
{
    private TextField inputArea = new TextField();
    private TextArea outputArea = new TextArea();
```

```

public static void main(String[] args)
{ Application.launch(args); }
@Override
public void start(Stage stage)
{
    Label headerLbl = new Label("Please insert Message in TextArea!"); Label
    inputLbl = new Label("Input: ");
    Label outputLbl = new Label("Output: ");
    Button okBtn = new Button("OK");
    HBox output = new HBox();
    output.getChildren().addAll(outputLbl, outputArea);

    okBtn.addEventHandler(MouseEvent.MOUSE_CLICKED,
    event -> outputArea.appendText("You: " + inputArea.getText() + "\n"));
    BorderPane root = new BorderPane();
    root.setTop(headerLbl);
    root.setRight(okBtn);
    root.setBottom(output);
    root.setLeft(inputLbl);
    root.setCenter(inputArea);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("SE2 Nachdenkzettel GUI");
    stage.show();
}
}

```

4.

Strukturen: Beschreiben Sie das Dokument einmal als Baum aus graphischen Nodes und einmal in Form eines serialisierten Textes (wie Html/xml). Die Tags dafür können Sie frei erfinden

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Label?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

```

```

<?import javafx.scene.text.Text?>

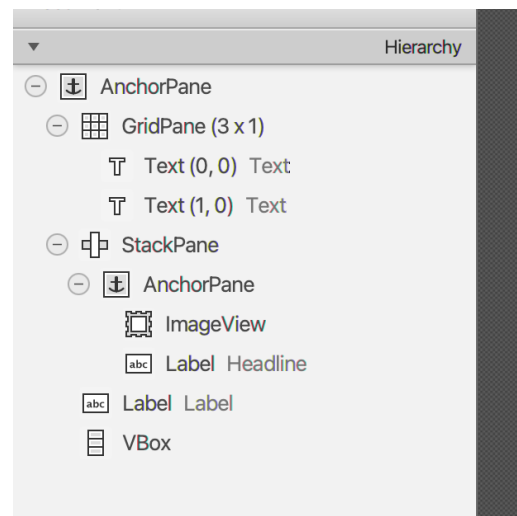
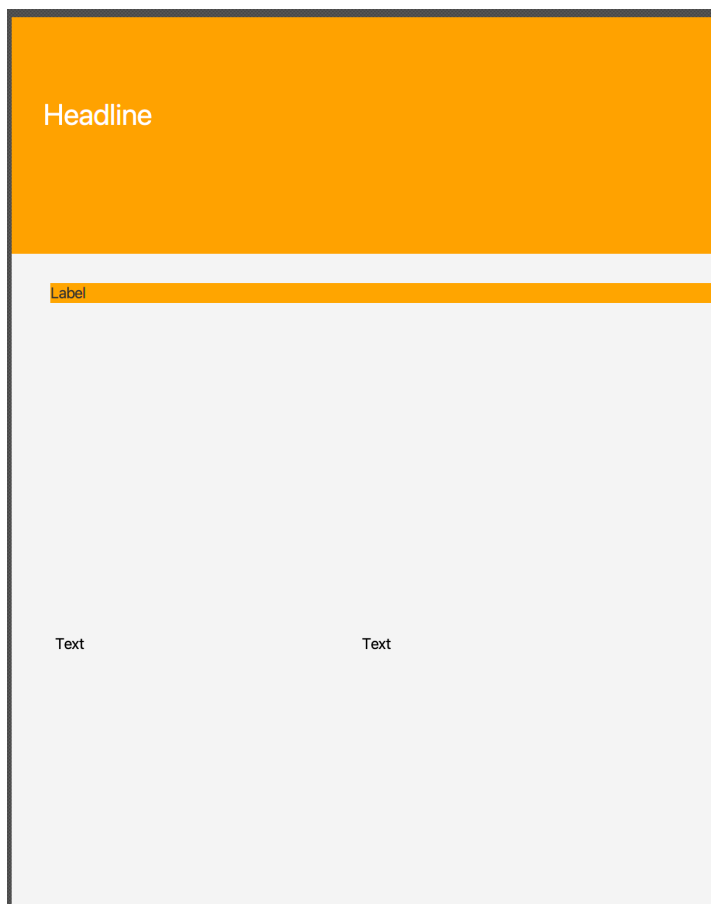
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="758.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/15.0.1" xmlns:fx="http://javafx.com/fxml/1">
    <children>
        <GridPane layoutX="37.0" layoutY="312.0" prefHeight="446.0" prefWidth="524.0">
            <columnConstraints>
                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
                <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
            </columnConstraints>
            <rowConstraints>
                <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
            </rowConstraints>
            <children>
                <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Text" />
                <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Text"
GridPane.columnIndex="1" GridPane.columnSpan="2" />
            </children>
        </GridPane>
        <StackPane prefHeight="105.0" prefWidth="600.0">
            <children>
                <AnchorPane prefHeight="184.0" prefWidth="600.0">
                    <children>
                        <ImageView fitHeight="202.0" fitWidth="600.0" pickOnBounds="true">
                            <image>
                                <Image url="@index.png" />
                            </image>
                        </ImageView>
                        <Label layoutX="27.0" layoutY="65.0" prefHeight="36.0"
prefWidth="157.0" text="Headline" textFill="WHITE">
                            <font>
                                <Font size="25.0" />
                            </font>
                        </Label>
                    </children>
                </AnchorPane>
            </children>
        </StackPane>
    </children>
</AnchorPane>

```

```

        </children>
    </StackPane>
    <Label layoutX="33.0" layoutY="227.0" prefHeight="17.0" prefWidth="567.0"
style="-fx-background-color: orange;" text="Label" />
    <VBox layoutY="205.0" prefHeight="553.0" prefWidth="600.0" />
</children>
</AnchorPane>

```



DocScape – publizieren mit Zukunft



6

+++ einfach in der Anwendung

DocScape bringt Ihre Kommunikationsprojekte nach vorne – individuell, innovativ, international

DocScape ist eine regelbasierte Software, die auf vorhandene Datenbanken oder Warenwirtschaftssysteme in Ihrem Unternehmen zugreift.

Ihr Corporate Design für Dokumente ist nichts anderes als eine umfangreiche und präzise Sammlung an Layoutregeln. DocScape übersetzt diese Gestaltungsregeln in eine XML-Notation und führt sie „auf Knopfdruck“ exakt und vollautomatisch aus.

Die Arbeit mit DocScape bedeutet: Änderungen einer Designregel werden in allen Dokumenten und Dokumenttypen automatisch umgesetzt. Auch ältere Versionen von Dokumenten werden von diesen Änderungen erfasst. Damit ist DocScape prädestiniert, die dynamische und nicht zuletzt durch die Expansion in neue Märkte getriebene Entwicklung eines Corporate Designs nachhaltig zu unterstützen. Die Arbeit mit DocScape verhindert darüber hinaus Flüchtigkeitsfehler oder das „bewusste“ Übersehen von CD-Regeln.

Sobald Sie „den Knopf“ drücken und DocScape starten, werden die relevanten Daten (Bilder, Grafiken, Texte, etc.) abgerufen und zu 100% gemäß dem hinterlegten Regelwerk auf den Seiten platziert – inklusive Schrift- und

Tabellenatz und bei Einhaltung aller definierten Stilvorgaben, in jeder beliebigen Sprache (wie z.B. Chinesisch, Arabisch, Japanisch oder Russisch). Ist etwa ein Bildständer gleich gewünscht, platziert DocScape die Bilder erst, nachdem alle relevanten Sprachen auf ihre Lauflänge hin überprüft und optimiert wurden. Innerhalb kürzester Zeit entsteht so das komplette Print-Produkt.

DocScape erzeugt auf einem gewöhnlichen Rechner (z.B. 1 GHz-Prozessor, 1 GB Speicher) etwa 50 bis 60 Seiten pro Minute! Das Programm arbeitet folglich so schnell, dass Sie selbst kurz vor der Drucklegung noch problemlos auch hochkomplexe Dokumente vollständig neu erzeugen können. Damit reduzieren Sie Fehlerquellen und Prozesskosten.

Mit DocScape ist auch die Produktion personalisierter Medien unkompliziert und wirtschaftlich möglich. Ob zielgruppenspezifischer Produktkatalog, individuelle Kundenzeitschrift oder persönliches Angebot – mit DocScape liefern Sie exakt diejenigen Informationen, die Ihre Zielgruppen oder Kunden wünschen. Besser lässt sich Kundenbindung nicht umsetzen. DocScape eröffnet Ihnen neue Dimensionen für das Direct-to-Customer-Marketing!

Sollen individuell gestaltete Seiten oder Objekte aus gängigen Anwendungen, wie z.B. InDesign, eingepflegt werden, ist auch dies problemlos möglich.

5. GUI Thread und andere...

a) Sie wollen Ihre JavaFX Application Unit Testen. Was für ein Problem tritt auf? Tipp: <https://medium.com/information-and-technology/test-driven-development-in-javafx-with-testfx-66a84cd561e0>

Es kann sein, dass es Probleme mit dem Anzeigen der verschiedenen Designelemente kommt. Z.B. die Layout Elemente überlappen, oder Textfelder werden übersprungen bei Tests.

b) Sie müssen Dinge im Background machen und können den Main GUI Thread nicht dafür nehmen? Sie müssen Daten zwischen GUI und Restapplikation austauschen? → Was macht die Task Class in JavaFX?

Tipp: <https://docs.oracle.com/javafx/2/threads/jfxpub-threads.htm>

Man sollte es über eine UI Thread (Java Application) Thread machen. Außerdem ist es hilfreich sich an dem `javafx.concurrent` package zu orientieren. So kann man multithreaded arbeiten, ohne viele Probleme mit den verschiedenen Threads zu haben.