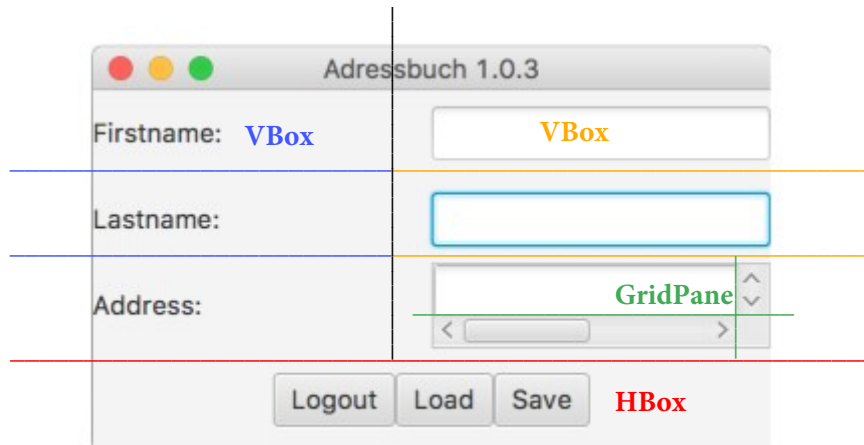


Nachdenkzettel zu GUI Programmierung

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.



2. Eventhandler

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

Answer:

- 1) **Foreground Events** = Events that require the direct interaction of a user
-> generated as consequences of a person interacting with graphical components in the GUI
- 2) **Background Events** = Events that don't require the direct interaction of a user
-> e.g.: hardware or software failure

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

Answer: Interface Callback

- e.g.: Pop up Windows to prompt for requested information

3. GUI: 1. Zeichnen Sie das Fenster mit Inhalt, das durch den JavaFX Code beschrieben wird. 2. Was macht der Handler?

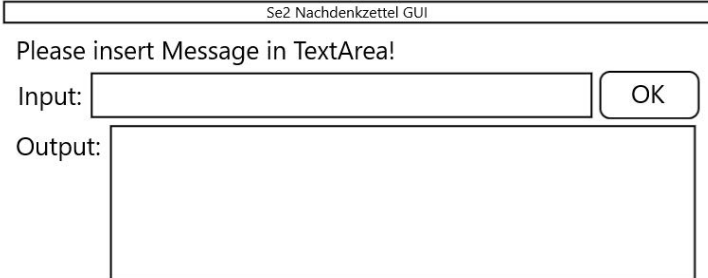
```
public class LayoutExample extends Application
{
    private TextField inputArea = new TextField();
    private TextArea outputArea = new TextArea();
    public static void main(String[] args)
    {
        Application.launch(args);
    }
    @Override
    public void start(Stage stage)
    {
        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();
    }
}
```

Answer:

1)



2) The Handler is triggered when the button (okBtn) is clicked. After that it prints the text a user typed in the TextField (inputArea) into the TextArea (outputArea).
-> e.g.: You: Hello my name is John.

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



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 Dokumententypen 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

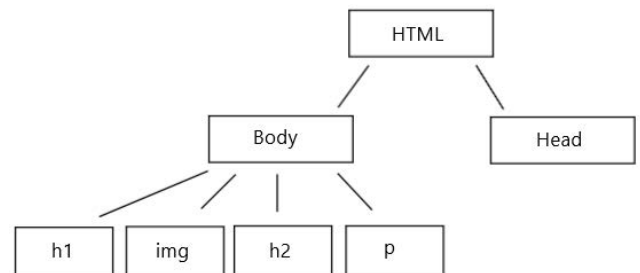
Tabellensatz und bei Einhaltung aller definierten Stilvorlagen, in jeder beliebigen Sprache (wie z.B. Chinesisch, Arabisch, Japanisch oder Russisch). Ist etwa ein Bildstandsabgleich gewünscht, platziert DocScape die Bilder erst, nachdem alle relevanten Sprachen auf ihre Lauflänge hin überprüft und optimiert 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.

```
<html>
  <head></head>
  <body>
    <h1>DocScape - publizieren mit Zukunft</h1>
    <img></img>
    <h2></h2>
    <p></p>
  </body>
</html>
```



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-java-fx-with-testfx-66a84cd561e0>

Answer: Challenges come from the lack of documentation by the TestFX developers

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>

Answer: A task class represents a single operation that doesn't return a value and usually executes asynchronous.