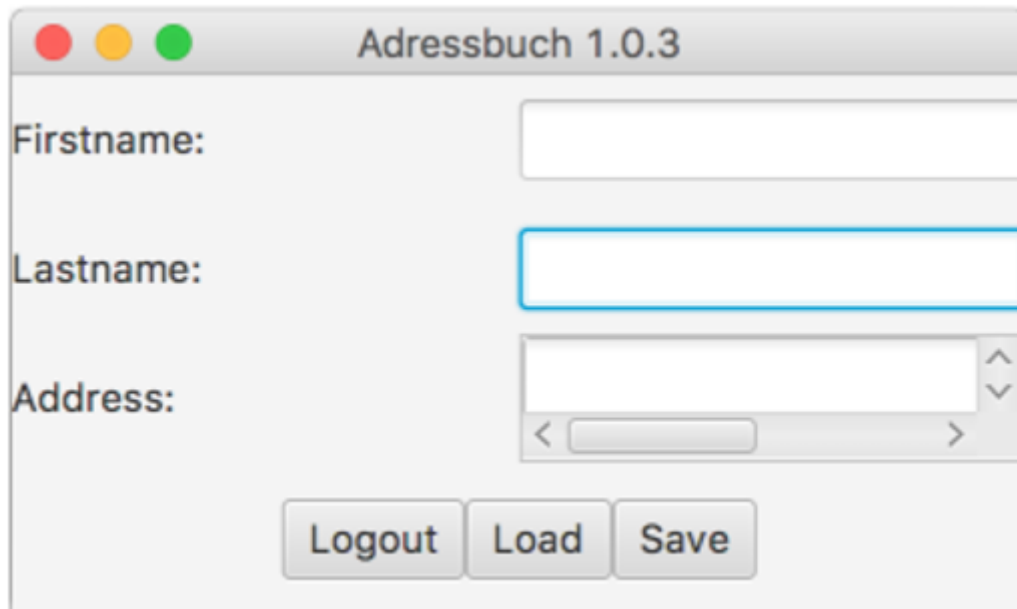


Nachdenkzettel: JavaFX

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



2. Geben Sie ein Beispiel für die zwei Formen von JavaFX Eventhandlern (alt und seit Java8).
3. Interpretieren Sie den folgenden Code

```
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();
}
}
```

- Zeichnen Sie das Fenster mit Inhalt, das durch den JavaFX Code beschrieben wird.
 - Was macht der Eventhandler?
4. Wie gehen Sie mit langlaufenden Operationen um, die durch einen Eventhandler gestartet werden (z.B. langlaufende Datenbank-Abfrage)? Was müssen Sie dabei beachten? Warum?

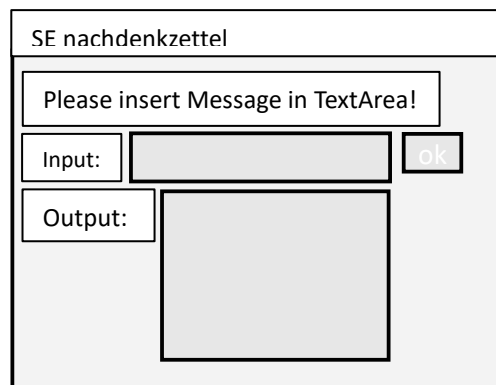
Nachdenkzettel: JavaFX

1. Ich würde eine Verschachtelung von mehreren bevorzugen, dass heißt:
BorderPane als Grundlage, dann rechts und links eine VBox und im unteren Bereich eine Hbox mittig-zentriert oder gepaart mit einem StackPane
2. Lambda-Ausdruck, `button.setOnAction(event -> System.out.println("Button clicked using newer event handler"));`

Ältere Methode: Anonymer Klassen-EventHandler

```
button.setOnAction(new javafx.event.EventHandler<javafx.event.ActionEvent>()
{
    @Override
    public void handle(javafx.event.ActionEvent event) {
        System.out.println("Button clicked using older event handler");
    }
});
```

3.



Wenn die Maus-clicked dann soll der Text: „You: (der Text von dem Input Eingabefeld) und ein Zeilenumbruch ausgegeben werden in dem Output Feld.

4. Die Operationen können längere Zeit beanspruchen auch wenn die nachfolgenden Operationen schon weiter gehen.
Um langlaufende Operationen asynchron auszuführen und den UI-Thread nicht zu blockieren, können Sie verschiedene Ansätze verwenden, einschließlich der Verwendung von JavaFX-eigenen Klassen und