

Frage: Wir haben unsere Aufgabenstellung Ihnen nur informell vorgestellt, müssen wir die noch präzisieren?

Antwort: Nein, in dieser Veranstaltung möchte ich nur grob das fachliche Thema und die eingesetzten Technologien wissen. Sie haben so in der Ausarbeitung einige Freiheitsgrade und können gerade mit der neuen Technologie experimentieren, wie es bei Forschungsansätzen üblich ist. Die präzise Aufgabenstellung (0.5 bis eine Seite) steht im ersten Kapitel der Hausarbeit. Wenn Sie die Aufgaben detaillierter besprechen wollen, können Sie natürlich auch eine kurze schriftliche Darstellung mit mir durchsprechen.

Frage: Sind sie außerhalb der Vorlesungszeit noch erreichbar?

Antwort: Ja. Generell antworte ich auf E-Mails, falls es nicht passiert, habe ich es schlicht vergessen, dann spätestens nach 4 Tagen nochmal schicken. Per Mail können wir auch einen Zoom-Termin vereinbaren, schicken Sie mir dann möglichst große Zeitintervalle, so dass ich dann einen Termin finden kann.

Frage: Wie kann ich JavaFX (OpenJFX) installieren?

Antwort: Da sich die Installation immer leicht verändert, der einfache Rat die KleukerSEU zu nutzen, da das dortige Java schon JavaFX enthält (sonst ein solches Java von einem Anbieter herunterladen). Es gibt so keinen GUI-Builder oder eine spezielle Projektversion, aber die wird generell nicht benötigt, darf aber natürlich auch genutzt werden. JavaFX-Oberflächen können mit einem GUI-Builder oder rein als Programmcode entwickelt werden. Beide Möglichkeiten werden Sie in der Hausarbeit in den Grundlagen erwähnen und dann, wenn Sie nicht JavaFX sehr detailliert analysieren wollen, eine auswählen. Keine der Auswahlen wirkt sich hier negativ auf die Bewertung aus. Beispiel:

```
package gui;

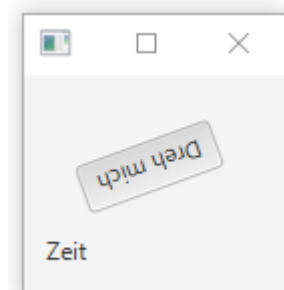
import java.time.Duration;
import java.time.LocalDateTime;
import javafx.application.Application;
import javafx.application.Platform;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;

public class Erste extends Application {

    private double value = 0;
    private boolean aktiv = true;
    private LocalDateTime start = LocalDateTime.now();

    @Override
    public void start(Stage primaryStage) {

        Label lbl = new Label("Zeit");
```



```

Button btn = new Button("Dreh mich");
btn.setOnAction(e -> {
    if (this.aktiv) {
        this.value += 40;
        btn.setRotate(value);
        if (this.value >= 360d) {
            this.aktiv = false;
            btn.setDisable(true);
            Duration dur = Duration.between(this.start
                , LocalTime.now());
            double zeit = dur.getSeconds() * 1e9 + dur.getNano();
            lbl.setText("" + zeit / 1e9);
        }
    }
});
lbl.setLayoutX(11);
lbl.setLayoutY(79);
btn.setLayoutX(27);
btn.setLayoutY(33);

AnchorPane root = new AnchorPane();
root.getChildren().add(btn);
root.getChildren().add(lbl);

Scene scene = new Scene(root, 130, 110);

primaryStage.setTitle("Dreher");
primaryStage.setScene(scene);
primaryStage.setOnCloseRequest(e -> Platform.exit());
primaryStage.show();
}

public static void main(String[] args) {
    Application.Launch(args);
}
}

```

Hinweis: Ein kleines Swing-Beispiel ist in den Folien zu MVC enthalten.

Frage: Undo-Möglichkeiten werden im Internet gerne mit einer undo-Methode() pro genutzter Aktion umgesetzt, ist das besser oder schlechter als der Praktikumsansatz in Blatt 10?

Antwort: Generell sind beide Versionen aus den Folien, mit undo()-Methode oder einem Undo-Objekt sinnvoll und nutzbar. Die Variante aus dem Praktikum wurde auch ausgewählt, um diese Möglichkeit der Aufteilung explizit zu zeigen. Weiterhin ist das redo() sehr einfach umsetzbar. Für die Praxis ist es „nur“ relevant, dass man frühzeitig daran denkt ein Undo einzuplanen, wenn es benötigt wird und dass man dazu sinnvolle Pattern kennt. Dabei kann auch ein einfaches Interface mit einer undo()-Methode ausreichen. Wir schauen uns nebenbei später die Redux-Architektur an, bei der ebenfalls recht einfach ein Undo möglich ist, man speichert Zustandskopien, ohne dass direkt ein Command-Pattern genutzt wird.

Generell ist eine Idee der Veranstaltung hier, dass man Möglichkeiten von Umsetzungen kennenlernt, also „seinen Werkzeugkoffer auffüllt“ und nicht meint, dass der erste Ansatz immer der passendste ist.