



GUI Übung 1 – Arbeitsfähigkeit

Beschreibung

Inhalt der Übung ist es, die nötige Software zu installieren und ein minimales Projekt mit JavaFX zu starten. Damit sind Sie für die kommenden Übungen arbeitsfähig.

Hinweis: Leider ist der Start in das erste JavaFX-Projekt spätestens seit Java 11 nicht mehr ganz intuitiv. Aber wir schaffen das. Bei Fragen/Problemen einfach melden.

Aufgabe 1 - Installation

Für die Übungen benötigen Sie:

- Aktuelles IntelliJ Ultimate (oder SEU nach Wahl)
- SceneBuilder for Java 11 and higher
- Git empfohlen

Stellen Sie sicher, dass auf Ihrem Arbeitsgerät diese Werkzeuge vorhanden und installiert sind.

Zusätzlich benötigen Sie ein aktuelles JDK (zum Beispiel OpenJDK), dies lässt sich allerdings auch über IntelliJ installieren.

Aufgabe 2 – das erste Projekt

Erzeugen Sie ein neues Gradle Projekt mit IntelliJ. Gehen Sie dabei wie folgt vor:

- Legen Sie ein neues Projekt vom Typ Gradle->Java an und wählen Sie eine JDK 17 aus (kann hier heruntergeladen werden).

Name: locationsFX

GroupId: de.throsenheim.gui

ArtifactId: locationsFX

- Passen sie den Gradle-Build an (Datei `build.gradle`)
 - Fügen Sie unter `plugins` das JavaFX- & das Application-Plugin hinzu
`id 'org.openjfx.javafxplugin' version '0.0.12'`
`id 'application'`
 - Ergänzen Sie auf oberster Ebene die Konfiguration für JavaFX

```
javafx {  
    version = "17"  
    modules = [ 'javafx.controls', 'javafx.fxml' ]  
}
```

- Geben Sie eine Startklasse an (wird im nächsten Schritt erzeugt), zum Beispiel:

```
mainClassName =
    'de.throsenheim.gui.locations.LocationsFxMain'
```
- Synchronisieren sie auf der rechten Seite im Tab Gradle das Projekt.
- Erzeugen Sie eine Main-Klasse mit entsprechendem Namen
- Füllen Sie die Klasse JavaFX-Code, zum Beispiel wie in „Listing 1“ unten im Anhang.
- Starten Sie das Projekt, indem sie im Tab „Gradle“ „Tasks -> application -> run“ ausführen
- Ergebnis: Ein einfacher Dialog startet und wird angezeigt.
Herzlichen Glückwunsch!

Aufgabe 3 – deklarative GUI mit FXML

Stellen Sie den Dialog auf FXML um und bearbeiten Sie ihn mit SceneBuilder. Gehen Sie dabei wie folgt vor:

- Erzeugen sie unter src/main/resources eine neue FXML-Datei „Hello.fxml“ **im gleichen Unterordner/-package wie die Klasse.**
- Entfernen sie ggf. die Zeile

```
„fx:controller=“de.throsenheim.gui.locations.Hello“
```
- Ersetzen sie den Inhalt der Klasse `LocationsFxMain` mit dem Beispiel aus „Listing 2“.
- Starten Sie die Anwendung erneut
- Ergebnis: Der neue Dialog wird angezeigt.
- Öffnen Sie die FXML-Datei mit SceneBuilder (Kontextmenu auf der Datei).
- Ergebnis: Die Oberfläche wird im SceneBuilder angezeigt und kann dort verändert werden.

Aufgabe 4 - GIT

Sie finden alle Übungen und Musterlösungen auf Github unter der folgenden URL:
<https://github.com/dominikhaas/Vorlesung-GUI-2022>

Clonen Sie sich das Repo in ein lokales Verzeichnis Ihrer Wahl

Diese Kopie sollten Sie einmal die Woche mit dem folgenden Kommando aktualisieren.
Damit bekommen Sie alle Übungsunterlagen direkt auf Ihren Rechner.

```
git pull
```

Anhang

Listing 1 – Simple application/dialog

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.stage.Stage;

public class LocationsFxMain extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        Label label = new Label("Hello world!");
        Scene scene = new Scene(label, 200, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

Listing 2 – FXML application

```
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class LocationsFxMain extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        Parent dialog =
FXMLLoader.load(this.getClass().getResource("Hello.fxml"));

        Scene scene = new Scene(dialog, 200, 200);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```