



# GUI Übung 3 – Events

## Beschreibung

Ziel der Übung ist es die Oberfläche aus Übung 2 mit Funktionalität zu versehen. Die Musterlösung für Übung 2 ist bei Github eingeecheckt.

Wenn der Button „Save“ betätigt wird, sollen alle Eingaben strukturiert ausgegeben werden.

## Aufgabe 1 – Event Handler in FXML

- Erstellen Sie, falls noch nicht vorhanden, eine Klasse `LocationFormController`. Hinterlegen Sie diese Klasse als Controller für die `LocationForm.fxml` in der IDE oder mit Scene Builder.
- Erstellen Sie im Controller eine Methode `onSave` als Event Handler mit passender Signatur.
- Registrieren Sie diese im Scene Builder für das Action-Event des Save-Button.

- Testen Sie, ob beim Betätigen des Buttons die Methode aufgerufen wird (Debugging oder Ausgabe auf der Konsole).

## Aufgabe 2 – Übernahme der Eingaben

- Schreiben Sie eine Klasse `LocationModel` als Pojo, die alle Werte des Dialogs (oder ausgewählte) übernehmen kann.  
Vorschlag für Datentypen: alles als `String`, `since` als `LocalDate` und `active` als `Boolean`
- Implementieren Sie im Controller eine Methode `private LocationModel toLocationModel()`, die ein neues `LocationModel`-Objekt erzeugt.
- Befüllen Sie in der Methode das neue Objekt mit den Werten aus den Eingabefeldern (mindestens `since`, `active` und zwei weitere Felder). Stellen Sie dazu eine Verbindung zwischen FXML und Code her.
- Beim Betätigen des Knopfes „Save“ soll das neue Objekt mit allen Attributen in der Konsole ausgegeben werden.

## Aufgabe 3 – programmatische Eventhandler

- Registrieren Sie den Event Handler für „Reset“ programmatisch, mit der einfachen Methode (`setOn...`).  
Beachten Sie dabei, an welcher Stelle im Code sie den Handler registrieren müssen. Die Methode selbst soll nur eine kurze Meldung auf der Konsole ausgeben.
- Registrieren Sie den Event Handler für „Cancel“ programmatisch mit der Rich-API. Die Methode selbst soll nur eine kurze Meldung auf der Konsole ausgeben.

## Aufgabe 4 – unterschiedliche Implementierungen

- Falls noch nicht passiert, setzen sie einen Event Handler als anonyme Klasse um.
- Falls noch nicht passiert, setzen Sie einen Event Handler als Lambda um.

## Aufgabe 5 (optional) – Event Filter

Verhindern Sie die Eingabe eines kleinen „a“ für alle Text-Felder (sinnfrei!). Gehen Sie dabei wie folgt vor:

- Injizieren Sie sich die Root Pane in den Controller.
- Registrieren Sie einen Event Filter für Key Events auf der Root Pane.
- Konsumieren Sie alle Key Events für „a“s.

Hinweise:

- Key-Event „`KEY_TYPED`“
- Feld `Character` im Event
- `consume()` konsumiert das Event