



GUI Übung 4 – MVx

Beschreibung

Ziel der Übung ist es, die Oberfläche aus Übung 3 nach dem MVP Muster mit Data Binding weiterzuentwickeln.

Aufgabe 1 – JavaFX Properties & Data Binding

Setzen Sie auf der Lösung von Übung 3 auf.

- Beantworten Sie für sich die folgende Frage: Was ist die Aufgabe der Klasse `EventModel`?
- Benennen Sie die Klasse `EventFormController` um zu `EventFormPresenter`.
- Ersetzen Sie alle Properties der Klasse `EventModel` durch JavaFX-Properties (`StringProperty`, `ObjectProperty<Integer>`, `ObjectProperty<LocalDate>...`)
- Erzeugen Sie getter/setter nach dem Muster der Vorlesung (`getter`, `setter` & `<name>Property()`)
- Entfernen Sie die `toSettings()` Methode aus dem Controller.
- Erzeugen Sie eine Instanz der Klasse `EventModel` im Controller und halten sie diese als Attribut.
- Stellen Sie in der `initialize()`-Methode des Controllers das Data Binding zwischen View und Model her. Änderungen an der GUI (Eingaben, Auswahl) sollen direkt in das Model geschrieben werden.
- Testen Sie ob die eingegebenen Werte korrekt im `EventModel` aktualisiert werden.

Aufgabe 2 – Listen & zusätzliche Bindings

- Erstellen Sie eine Klasse `EventFormModel`, welche die Auswahlmöglichkeiten der Combo-Boxen Start & End der GUI kapselt. Die auswählbare Werte sind 0-23.
Hinweis: Verwenden Sie die `new SimpleListProperty<>(FXCollections.observableArrayList())` um ein passendes Property zu erzeugen.
- Verbinden Sie diese Klasse ebenfalls per Data Binding mit dem Dialog und den Auswahllisten für „Start“ & „End“.

Aufgabe 3 – Change Listener

Setzen Sie bei der Auswahl eines neuen Werts im Feld „Start“ die verfügbaren Werte in End neu. End darf nicht vor Start liegen.

Hinweise:

- Binden sie dazu die Auswahl von „Start“ zusätzlich auch in ein Property im `EventFormModel`.
- Aktualisieren Sie die Listen dadurch, dass Sie `.set(..)` auf der `ListProperty` aufrufen.