

Selbststudiums-Aufgabe 12

Presentation Model, MVVM

1. Adder mit Presentation Model oder MVVM

Im Gegensatz zur .NET-Technologie können Sie bei JavaFX nicht einfach die Controls in der *.fxml*-Datei deklarativ an Properties anbinden, sondern es braucht weiterhin eine Klasse, in der Variablen entsprechender Typen mit der *@FML*-Annotation deklariert sind. Dort können auch die Bindings an Properties durch entsprechenden Java-Code programmiert werden.

Will man dem *Presentation Model*-Muster oder dem *MVVM*-Muster folgen, braucht es in JavaFX daher immer noch auch eine Code-Klasse für die View.

Programmieren Sie eine Variante des *Adder*-Programms aus Selbststudium 11 nach dem *MVVM*-Muster. Für das *Model* können Sie gerade die *Model*-Klasse ohne JavaFX-Abhängigkeit aus der entsprechenden MVC-Lösungsvariante verwenden. In Ihrer *MVVM*-Lösung sollten nur *ViewModel* und *View* JavaFX-spezifisch sein. Die *View*-Klasse sollte keinerlei Code ausser für *Bindings* enthalten.

Für diese Aufgabe können Sie die *.fxml*-Datei unverändert von vorherigen Lösungen übernehmen. Sollten Sie allerdings den Namen der Klasse mit den entsprechenden Controls ändern, müssen Sie das auch in der *.fxml*-Datei entsprechend eintragen. (Nebenbei: Das entsprechende XML-Attribut heisst *fx:controller*. Finden Sie diesen Namen eigentlich gut?)

2. Control Enabling

Für diese Aufgabe brauchen Sie nichts zu programmieren, sondern nur zu entscheiden, in welchen Klassen Sie was einfügen würden.

Stellen Sie sich vor, das *Adder*-Programm sollte so abgeändert werden, dass die Summe nicht mehr automatisch bei jeder Änderung eines der Summanden berechnet wird, sondern erst auf expliziten Befehl hin. Das *Model* werde dazu um eine zusätzliche Methode *add()* erweitert, die im Bedarfsfall aufgerufen werden kann.

Im User Interface soll ein *Button Control* zusätzlich angezeigt werden. Dieser *Button* soll *disabled* („ausgegraut“) sein, wenn die Summe seit der letzten Neuberechnung noch aktuell ist und noch keiner der Summanden wieder verändert wurde. Erst wenn in einem der Summanden-Felder ein neuer Wert eingetippt wird, soll der *Button* wieder *enabled* werden, bis er das nächste Mal gedrückt wird.

Das *Button Control* in JavaFX gibt über eine Methode *BooleanProperty disableProperty()* Zugang zu einer Property mit der gesteuert werden kann, ob der *Button* gerade *enabled* oder *disabled* ist. Diese Property kann auch an eine beliebige andere *BooleanProperty* gebunden werden.

In welchen Klassen

- i) deklarieren Sie das Button-Control
- ii) programmieren Sie den Aufruf der *add()*-Methode im *Model*
- iii) programmieren Sie Code, der entscheidet, ob der *Button* jeweils *enabled* oder *disabled* sein soll

wenn Sie

- a) dem *MVC*-Muster folgen?
- b) dem *Presentation Model* bzw. *MVVM*-Muster folgen?

Abgabe bis 18. Juni 2018 bzw. 19. Juni 2018