

OOP1 Dokumentation

Referenzverwaltung

Rafael Stauffer

4. Juli 2022

Inhaltsverzeichnis

0.1	Versionierung	3
0.2	Funktionen	3
0.2.1	Funktion1 Objektklassen	3
0.2.2	Funktion 2 JavaFX GUI	4

0.1 Versionierung

<i>Datum</i>	<i>Name</i>	<i>Beschreibung</i>
03.07.2022	Stauffer	Initiale Version aus Notizen

0.2 Funktionen

0.2.1 Funktion 1 Im Minimum eine Objekt Klasse erstellt.

Anforderung

In der Konzept-Phase beim Szenarienbeschrieb wurden drei Business-Object-Klassen identifiziert:

- Buch
- Referenz
- Regelbereich

Design

Durch das Proof of Concept für die BibTex-Dokumente wurden die Business-Object-Klassen mit der Unterscheidung von Buch und Website auf vier erweitert.

Ein weiteres Normalisieren der Daten auf Author wurde wegen dem Projektumfang verworfen. Interfaces zu den in der Anforderung definierten Objekten wurden für die zukünftige Flexibilität miteinbezogen. Sämtliche Attribute wurden zur Einfachheit als Strings festgelegt.

- BuchSource
- WebseitenSource
- Referenz
- Regelbereich

Entwicklung

Umsetzung wie nach Design. Erste Version nur Referenz mit Source und Regelbereich als Dummy-Objekt zum Testen bevor die vollständige Funktionalität umgesetzt wurde. Nach dem ersten Prototypen wurden Einschränkungen der 'jedes Attribut ein String' Methodik sichtbar. Zur besseren Überprüfung der Eingaben und Zukunftssicherheit wurde das Erscheinungsjahr/LetzterAufruf-Feld als LocalDate definiert welches einfacher dem

DatePicker übergeben werden kann. Website wurde als URL definiert was kurzzeitig für Probleme bei der Übergabe an die View gesorgt hat aber ungültige URLs automatisch erkennt. Ursprünglich wurde eine Persistenz der Daten eingeplant aber wegen dem Projektumfang verworfen und stattdessen das DataAccessObject-Pattern eingebaut um diese zukünftig nachzuliefern.

Test

Zusammen mit Funktion 3 getestet. Alle Objekte wurden wiederholt erstellt, geändert und gelöscht.

Fazit

Die Vererbung und Interfaces könnten eleganter gelöst sein. Der bisherige Aufbau stellt ein Kompromiss zwischen Aufwand/Projektumfang und Erweiterbarkeit dar.

0.2.2 Funktion 2 JavaFX als grafische Oberfläche eingesetzt mit mindestens 2 Views

Anforderung

Der Szenarienbeschrieb beinhaltet drei Paare mit je einem Übersichts- und einem Detail-Screen im Total sechs Views.

Design

Für die Übersichts-Screens wurden ListView geplant um eine einfache Ansicht in einem minimalistischen Design bereitzustellen mit Änderungs und Löschfunktionen direkt auf der Zeile. Die Detail-Screens wurden als Erstellungs- und Bearbeitungs-Screen geplant welche je nach Aufruf Nodes ein- oder ausblenden aufgebaut auf VBox/HBox und Text-boxen. Es wurde noch nicht festgelegt ob die Detail-Screens in einem neuen Fenster/Stage erscheinen oder die aktuelle Stage übernehmen sollten.

Entwicklung

Beim der Entwicklung des Referenz-Screen-Paars wurde klar, dass:

- die ListView zu unflexibel ist für die verschiedenen Business-Objekte: TableView stattdessen verwendet
- VBox/HBox-Verschachtelungen die Detail-Screens nur bedingt darstellen können: GridPanes wurden stattdessen verwendet
- die Stage übernahme des Detail-Screens vom Übersichtscreens die komplexität erhöht ohne grosse Vorteile: Detail-Views starten somit in eigener Stage
- die Views immer etwa gleich aufgebaut sind: eine BaseController-Klasse wurde eingeführt

-
- SubViews die Komplexität massiv erhöhen: keine SubScreens verwendet

Obwohl weitere Entkoppelungen der Nodes besonders im Overview-Screen möglich sind wurden diese wegen Zeitdruck nicht umgesetzt.

Test

- Starten und Beenden der Views in eigenen Stages.
- Starten und Beenden der Views in Parent Stage.
- Parameterübergabe zwischen Controller.
- Aufruf von Events.
- Verändern von View-Feldern.
- Vererbung von Event und Feldverknüpfungen

Fazit

Die Arbeit mit FXML war besonders ermüdend, da kaum Debugginginformationen weitergereicht wurden.