

Übungsblatt 4

Aufgabe 1: Vertiefungsaufgabe Entwurfsmuster Befehl (BL)

Lesen Sie die Beschreibung des Entwurfsmusters Befehl in J. Goll, S. 153 ff.

- a) Implementieren Sie das Beispiel aus Bild 4-32, „Klassendiagramm zum Programmbeispiel Befehlsmuster“, folgen Sie dabei den Beschreibungen aus der Beschreibung von Goll.
- b) Erstellen Sie ein UML-Klassendiagramm zu dem Beispiel aus a).
- c) Nun soll Ihr Beispiel aus a) erweitert werden, um einen weiteren Befehl „LichtDimmen“ umzusetzen. Wo müssen Sie Änderungen durchführen? Integrieren Sie den Befehl „LichtDimmen“ in Ihr Programm.
- d) Ihr Beispiel soll nun so erweitert werden, dass eine Befehlshistorie realisiert wird, d.h. nach Ausführung eines Befehls soll ein Befehl mit einer Ausführungszeit versehen werden und gespeichert werden.

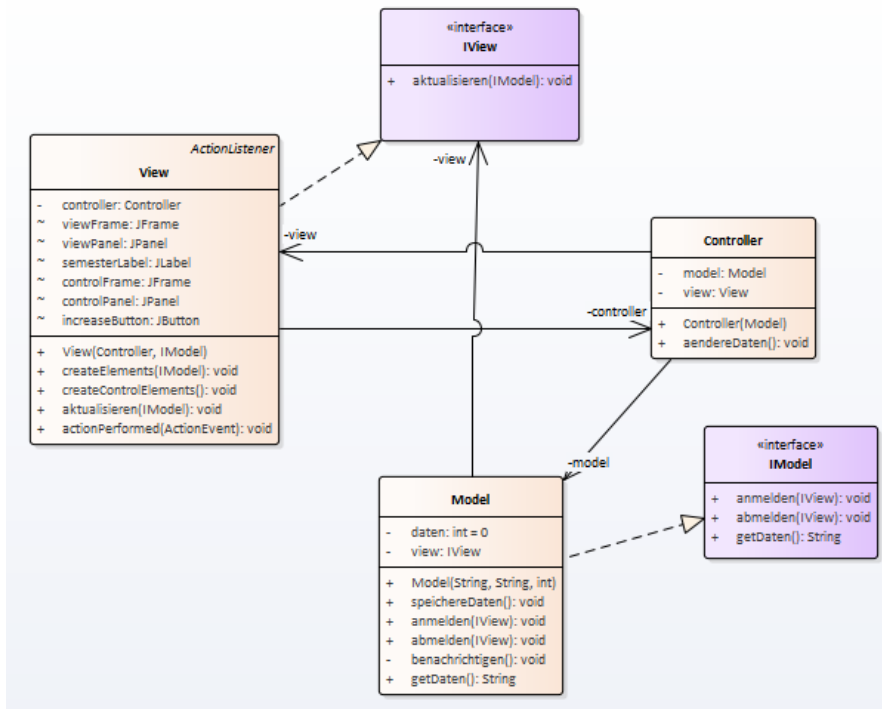
Dazu erstellen Sie eine weitere Klasse Befehlshistorie. Verwenden Sie für die Speicherung der Befehle einen Stack.

Mit Hilfe der Befehlshistorie soll eine „Undo“-Funktionalität realisiert werden, mit der ein Befehl rückgängig gemacht werden kann. Dazu erweitern Sie das Interface IBefehl mit der Methode undo und setzen erforderliche Änderungen an den Befehlsklassen um.

Schreiben Sie ein Testprogramm, um die neue Funktionalität zu demonstrieren.

Software Engineering
Wintersemester 2022/2023
Prof. Dr. Jochen Küster

Aufgabe 2: Model-View-Controller



- Gegeben ist die Struktur des MVC-Musters sowie ein erster Entwurf für die Umsetzung in UML. Wo finden Sie im Entwurf das Beobachter-Entwurfsmuster? Identifizieren Sie für jede Klasse des Beobachter-Entwurfsmusters die entsprechende Klasse im obigen Entwurf.
- Entwickeln Sie auf Basis des Entwurfs ein Java Programm, welches den Model-View-Controller Ansatz anwendet. Dazu ist auch eine erste Implementierung vorgegeben. Es sollen Daten eines Studierenden (repräsentiert als `StudentModel`) angezeigt werden und die Semesteranzahl des Studierenden kann über einen einfachen Knopf (Button) erhöht werden. Vervollständigen Sie das Programm, so dass sinnvolle Daten des Studierenden angezeigt werden.
- Zeichnen Sie ein Sequenzdiagramm, das die Interaktion darstellt, wenn bei `StudentView` der Knopf gedrückt wird.
- Erweitern Sie Ihr Programm um eine weitere sinnvolle View, z.B. zur Anzeige weiterer Daten, die nur für die Lehrer/Hochschullehrer sichtbar sind.

Software Engineering
Wintersemester 2022/2023
Prof. Dr. Jochen Küster

Aufgabe 3: MVC untersuchen und umsetzen

Entwickeln Sie das Programmbeispiel aus J. Goll (Abschnitt 5.6.7) zu einem lauffähigen Java Programm. Benutzen Sie dazu zunächst den vorgegebenen Code.

- a) Untersuchen Sie den Code. In diesem Fall gibt es im Code eine wichtige Abweichung von der Standardimplementierung eines Beobachter-Musters, die mit den Abhängigkeiten zu tun hat. Welche?
- b) Versuchen Sie, den Code so zu verändern, dass er der Standardimplementierung des Beobachter-Musters folgt.
- c) Fügen Sie eine weitere View Ihrer Wahl hinzu.

Aufgabe 4: Schichtenarchitektur entwickeln

Entwickeln Sie das Programmbeispiel aus J. Goll (Abschnitt 5.6.7) weiter, in dem Sie eine weitere Schicht, die Datenhaltungsschicht entwickeln.

- a) Entwerfen Sie Klassen für das Schreiben/Lesen von Daten aus einer Datenbank. Implementieren Sie diese Klassen in Java.
- b) Erstellen Sie eine Verbindung zwischen dem Programmbeispiel von J. Goll und Ihren neuen Klassen zur Datenhaltung. Vermeiden Sie dabei unnötige Abhängigkeiten zwischen den Klassen. Machen Sie die Anwendung lauffähig, so dass Daten geladen/gespeichert werden können
- c) Erstellen Sie ein UML-Klassendiagramm Ihrer Anwendung und ordnen Sie die Klassen verschiedenen Schichten zu.