

# Software Engineering Übung 11:

## Gruppe:

1. Michael Scholz (matr.: 1576630, rbg: mi48azih)
2. Ulf Gebhardt (matr.: 1574373, rbg: hu56nifa)
3. Sebastian Weicker (matr.: 1625099, rbg: we87obyq)

## Aufgabe 1: Anwendung von Observer Pattern:

a)

→ siehe Code

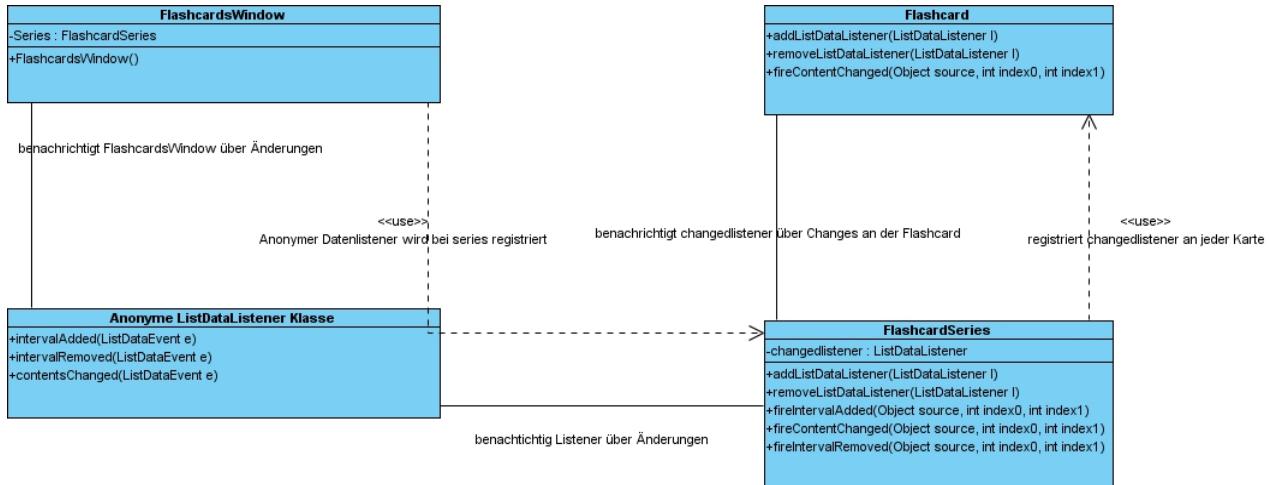
b)

→ siehe Code

c)

→ siehe Code

d)



→ Größere Darstellung im beigelegten EX11 A1.4.jpeg

Klasse	Rolle im Pattern
FlashcardsWindow	Keine
Flashcard	Subjekt
FlashcardSeries	Subjekt
Anonyme ListDataListener	Beobachter

e)

Es gibt zwei Möglichkeiten, wer ein update auslöst:

1. Die ändernde Methode
2. Das geänderte Objekt selbst

In unserer Implementierung haben wir das geänderte Objekt als denjenigen gewählt, der das Update auslöst. In unserem Fall ist das FlashcardSeries für neue oder entfernte Karten und Flashcard selbst für Änderungen innerhalb einer Flashcard.

**Vorteil** dieser Methode: Es wird sichergestellt, dass wirklich alle Änderungen als Update an die Listener weitergegeben werden. Außerdem muss derjenige, der das Objekt ändert sich nicht darum kümmern, den Aufruf von update() zu machen.

**Nachteil:** Es werden gegenfalls mehrere Updates getriggert, was zu wiederholten Aktualisierung der FlashcardSeries-Anzeige führen kann.

f)

Wir benutzen die Push-Methode. Dabei wird das geänderte Objekt mit dem Event and alle Observer geschickt. Bei Änderungen in der FlashcardSeries werden beim Löschen von Flashcards auch deren indizes übergeben.

Bsp :Änderung in einer Flashcard:

1. Flashcard wird geändert
2. Flashcard ruft fireContentChanged(this,0,0) auf
3. fireContentChanged erstellt ein ListDataEvent, welches die Referenz zu der geänderten Flashcard enthält und gibt dieses weiter an alle registrierten Observer.