

## Übung 2

Erstellen Sie das CLI für den Beleg und die beiden Beobachter.

Eine prototypische Lösung würde 4 Befehle (und keine Zustände) kennen: ein Befehl zum Anlegen eines Kunden, zwei Befehle zum Einfügen zweier verschiedener (vordefinierter) Frachtstücke und ein Befehl zum Löschen eines Frachtstücks.

Der Persistenzmodus und der Konfigurationsmodus sowie die Ausführung als Client bzw. Server sind nicht Teil der Übung.

### Abgabeanforderungen

Die Abgabe hat als zip-Datei zu erfolgen, die ein lauffähiges IntelliJ-IDEA-Projekt enthält. Sie muss eine (in IntelliJ lesbare) readme-Datei im root des Projektes (neben der iml-Datei) enthalten in der der erreichte Stand (die Nummer) bezüglich des Bewertungsschemas vermerkt ist.

### Quellen

Zulässige Quellen sind suchmaschinen-indizierte Internetseiten. Werden mehr als drei zusammenhängende Anweisungen übernommen ist die Quelle in den Kommentaren anzugeben. Ausgeschlossen sind Quellen, die auch als Beleg oder Übungsaufgabe abgegeben werden oder wurden. Zulässig sind außerdem die über moodle bereitgestellten Materialien, diese können für die Übungsaufgaben ohne Quellenangabe verwendet werden.

### ~~in der Übung vorzuführen~~

`Counter` mit events und Beobachterentwurfsmuster implementieren, sodass die Zustandsübergänge auf der Konsole dargestellt werden und die vorgegebenen Tests erfolgreich durchlaufen

## Bewertungsschema

### 1 Punkt:

1. zip Archiv
2. IntelliJ-Projekt (kein Gradle, Maven o.ä.)
3. readme vorhanden
4. Java8 (language level)
5. kompilierbar
6. main-Methoden nur im default package
7. ausführbar
8. Darstellungslogik und Geschäftslogik getrennt
9. prototypisches CLI (nicht notwendig, wenn umfangreicheres CLI realisiert ist)

### 2 Punkte:

- ~~10. Präsenzaufgaben erfüllt~~
11. Trennung zwischen Test- und Produktiv-Code
12. JUnit5 als Testframework
13. Mockito als Mock/Spy-framework
14. keine leeren Test
15. Beobachterentwurfsmuster und events realisiert
16. beide Beobachter getestet
17. zwei listener getestet

### 2,5 Punkte:

18. angemessene Aufzählungstypen verwendet
19. nach MVC strukturiert