

# Übung 4

Erstellen Sie die graphische Oberfläche für den Beleg.

## 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.

## fakultativ

JavaFX-Anwendung mit 2 Texteingabefeldern und zwei buttons, der linke button kopiert den Text aus dem linken Texteingabefeld in das rechte, der rechte button spiegelt dieses Verhalten

1. imperativ programmierte View
2. deklarativ programmierte View

## 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. Einfügen, Auflisten und Löschen von Frachtstücken in der GUI

### 2 Punkte:

9. Trennung zwischen Test- und Produktiv-Code
10. JUnit5 als Testframework
11. Mockito als Mock/Spy-framework
12. keine leeren Test
13. Geschäfts und Darstellungslogik getrennt
14. View und Model synchronisiert
15. sortierbare Darstellung der Fracht mit Lagerposition, Einlagerungsdauer und Kunde
16. skalierbare Darstellung

### 2,5 Punkte:

17. Änderung der Lagerposition mittels drag&drop
18. data binding verwendet