

Software Engineering

## Praktikumsversuch 1 Gruppe E - Hausaufgaben -

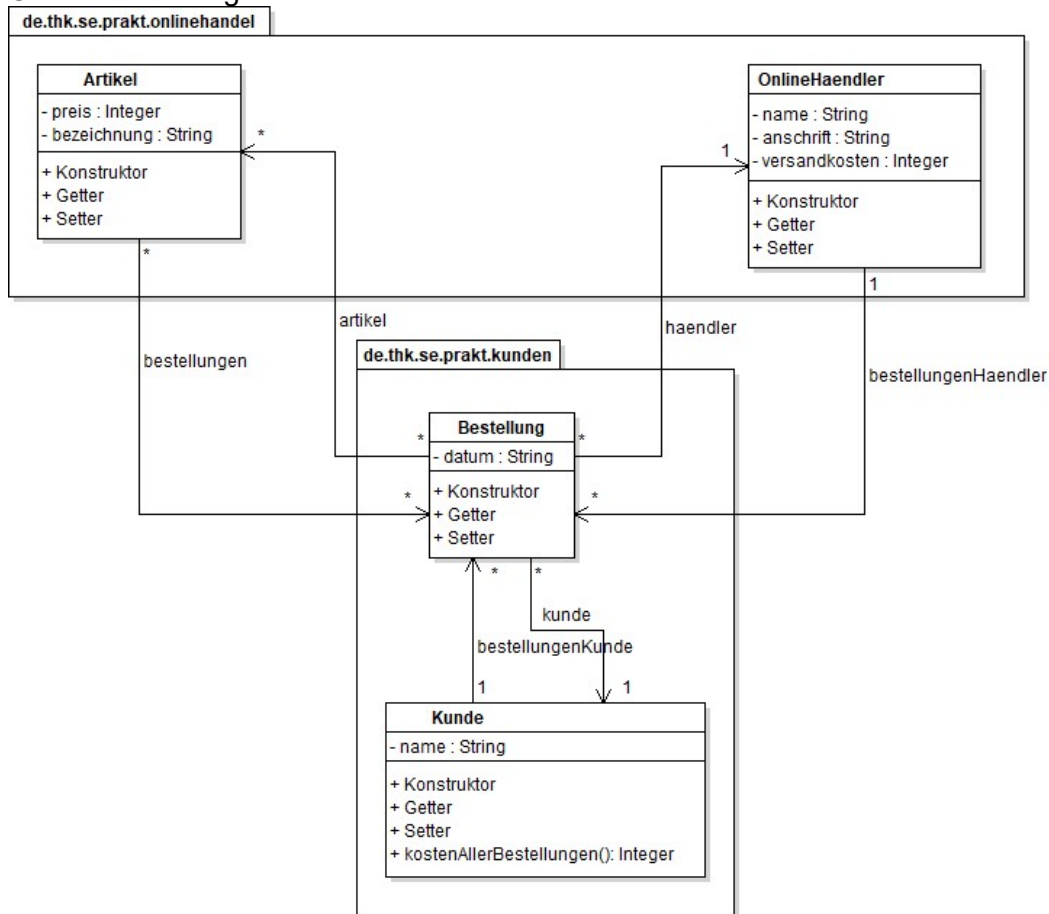
**Ziel:** Implementierung eines UML-Klassendiagramms, Sequenzdiagramm für eine Methode

**Abgabe der Lösungen:** Bis zum 16.11., 08:00 Uhr morgens, im Master-Branch Ihres Gitlab-Repositories P1SE<IhreTeilnehmernummer> (z.B. <https://gitlab.nt.fh-koeln.de/gitlab/se/SE29/P1SE29.git> für Teilnehmer 29). Abzugeben ist das vollständige IntelliJ-Projekt und das Sequenzdiagramm. Das Sequenzdiagramm muss das Format pdf oder jpg besitzen und im Ordner docs im IntelliJ-Projekt abgegeben werden.

In Ihrem Gitlab-Repository zu diesem Praktikumsversuch habe ich Ihnen ein IntelliJ-Projekt zur Verfügung gestellt, welches Sie als Grundlage Ihrer Implementierung verwenden müssen. Weiterhin befindet sich in dem IntelliJ-Projekt der Ordner docs, in dem Sie das erstellte Sequenzdiagramm abgeben müssen.

**Gegeben:**

UML-Klassendiagramm:



Das Klassendiagramm stellt die Verwaltung von Bestellungen im Online-Handel dar. Ein Online-Händler besitzt einen Namen und eine Anschrift. Für jede Bestellung berechnet ein Online-Händler bestimmte Versandkosten. Ein Artikel besitzt eine Bezeichnung und einen Preis. Ein Kunde besitzt einen Namen. Ein Kunde kann viele Bestellungen. Jede Bestellung gehört zu genau einem Kunden und einem Online-Händler. Eine Bestellung besitzt ein Datum und umfasst beliebig viele Artikel (Jeder Artikel aber immer nur in Anzahl = 1).

## **Aufgaben:**

### **A 1.1 UML-Klassendiagramm in Java implementieren**

Erstellen Sie Java-Klassen entsprechend des obigen UML-Klassendiagramms.

Verwenden Sie IntelliJ IDEA zur Implementierung.

Implementieren Sie für alle Attribute des Klassendiagramms entsprechende Getter und Setter.

Implementieren Sie für jede Klasse genau einen Konstruktor, der die Vorgaben des Klassendiagramms korrekt umsetzt.

Implementieren Sie für mehrwertige Assoziationen Verwaltungsoperationen zum Hinzufügen und zum Entfernen von Objekten.

Alle Attribute der Java-Klassen müssen `private` sein.

Implementieren Sie die Methode `kostenAllerBestellungen()` an der Klasse `Kunde`. Diese Methode soll für eine Kunde die Kosten aller seiner Bestellungen berechnen. Die Kosten einer Bestellung setzt sich zusammen aus den Versandkosten des zugehörigen Online-Händlers und den Preisen aller Artikel dieser Bestellung. Sie können hierzu beliebige weitere Methoden an den anderen Klassen erzeugen.

### **A 1.2 Javadoc erstellen**

Erstellen Sie für die Klasse `Kunde` und der dort vorhandenen Methoden aussagekräftige Javadoc Kommentare. Die HTML-Seiten zu Javadoc müssen Sie nicht generieren lassen.

### **A 1.3 Sequenzdiagramm erstellen**

Erstellen Sie für die von Ihnen implementierte Methode `kostenAllerBestellungen()` ein Sequenzdiagramm. In dem Sequenzdiagramm müssen dann auch die Abläufe der von Ihnen ggfs. zusätzlich erstellten Methoden modelliert werden.

Verwenden Sie im Sequenzdiagramm die gleichen Bezeichnungen für Variablen, Parameter usw. wie in Ihrem Code.

Sie können das Diagramm mit draw.io oder einem anderen Werkzeug erstellen, oder auch per Hand erstellen und dann einscannen. Sie müssen das Sequenzdiagramm in allen Fällen im Format pdf oder jpg abgeben.

### **A 1.4 Objekte erzeugen und Methode ausführen:**

Erstellen Sie eine Klasse `OnlineHandelTest` mit einer main-Funktion im Paket `de.thk.se.prakt.test`. In der main-Funktion erstellen Sie für die Java-Klassen aus Aufgabe A 1.1 die folgenden Objekte. Achten Sie darauf, dass Sie alle im Klassendiagramm geforderten Assoziationen für die Objekte erstellen.

Online-Händler:

- Händler1: Name: Schmidt, Anschrift: Köln, Versandkosten: 18
- Händler2: Name: Jeff, Anschrift: Düsseldorf, Versandkosten: 56
- Händler3: Name: OnlineParadise, Anschrift: Aachen, Versandkosten: 31

Kunde:

- Name: Peter
- Name: Klaus
- Name: Martin

Artikel:

- Artikel1: Bezeichnung: Bohrmaschine, Preis: 206
- Artikel2: Bezeichnung: Schraubenset, Preis: 13
- Artikel3: Bezeichnung: Druckerpatrone, Preis: 104
- Artikel4: Bezeichnung: Buch, Preis: 22
- Artikel5: Bezeichnung: Rasenmäher, Preis: 390

Bestellungen:

- Bestellung1: Datum: „13.08.2021“, Kunde: Martin, Online-Händler: Händler3, Artikel: Artikel3, Artikel1, Artikel5
- Bestellung2: Datum: „01.09.2020“, Kunde: Peter, Online-Händler: Händler2, Artikel: Artikel3, Artikel4, Artikel5
- Bestellung3: Datum: „16.10.2021“, Kunde: Klaus, Online-Händler: Händler3, Artikel: Artikel4, Artikel2, Artikel3
- Bestellung4: Datum: „22.04.2020“, Kunde: Martin, Online-Händler: Händler1, Artikel: Artikel1, Artikel2, Artikel3

Führen Sie dann für alle Kunden separat die Operation `kostenAllerBestellungen()` aus und geben Sie das Ergebnis in lesbarer Form auf der Konsole aus.