

Software Engineering

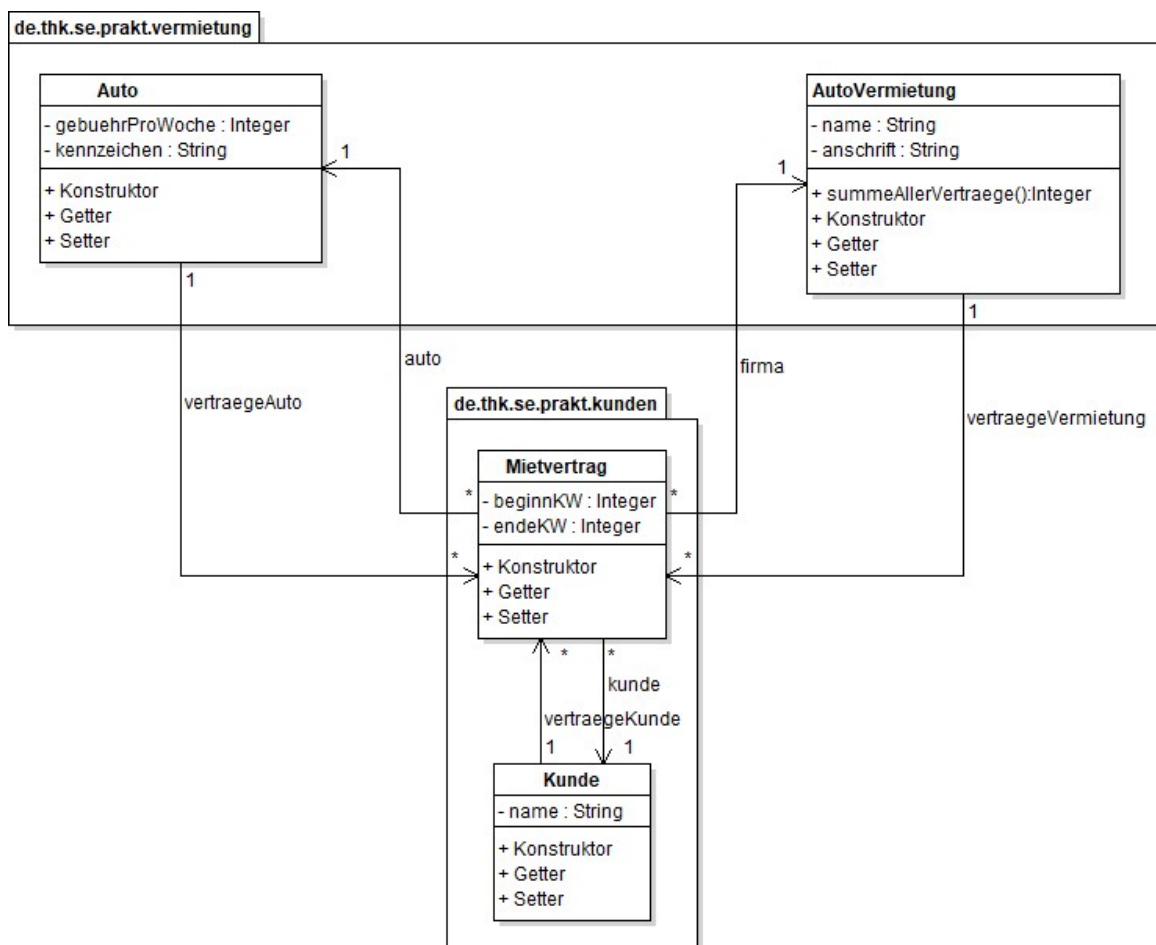
## Praktikumsversuch 1 Gruppe A - Hausaufgaben -

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

**Abgabe der Lösungen:** Bis zum 09.11., 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:



Es stellt eine Autovermietung dar, die Autos an Kunden vermietet. Ein Mietvertrag wird zwischen einem Kunden und einer Auto-Vermietung für ein bestimmtes Auto abgeschlossen. Ein Auto kann immer nur wochenweise gemietet werden und nur mindestens für eine Woche. Die Mietgebühren

sind für ein Auto pro Woche angegeben (`gebuehrProWoche`). Für den Mietvertrag sind die Kalenderwoche des Beginns (`beginnKW`) und des Endes (`endeKW`) des Mietvertrages registriert. Bei Vermietung eines Autos für nur eine Woche, besitzen `beginnKW` und `endeKW` den gleichen Wert. Die Vermietung über die Jahresgrenze hinaus ist möglich, d.h. in diesem Fall besitzt `beginnKW` einen höheren Wert als `endeKW`. Ein Auto wird nie länger als 51 Wochen vermietet.

## 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 `summeAllerVertraege()` an der Klasse `AutoVermietung`. Diese Methode soll für eine Autovermietung die Summe der Mietpreise aller vorhandenen Mietverträge ermitteln. Sie können hierzu beliebige weitere Methoden an den anderen Klassen erzeugen.

Hinweis: Ein Jahr hat hierbei immer genau 52 Kalenderwochen!

### A 1.2 Javadoc erstellen

Erstellen Sie für die Klasse `AutoVermietung` 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 `summeAllerVertraege()` 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 `AutoTest` 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.

Auto:

- Auto1: Kennzeichen: AC-P 1234, 20 € pro Woche
- Auto2: Kennzeichen: DN-T 2345, 30 € pro Woche
- Auto3: Kennzeichen: K-TZ 9876, 123 € pro Woche

Kunde:

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

Autovermietung:

- Name: Minipreis, Anschrift: Koeln
- Name: Maxipreis, Anschrift: Bonn

Mietvertrag:

- Vertrag 1: Minipreis - Peter – Auto1 – KW 2 bis KW 3
- Vertrag 2: Minipreis - Martin – Auto3 – KW 51 bis KW 3
- Vertrag 3: Maxipreis - Peter – Auto2 – KW 4 bis KW 4
- Vertrag 4: Maxipreis - Ernst - Auto2 – KW 10 bis KW 22

Führen Sie dann für beide Autovermietungen separat die Operation `summeAllerVertraege()` aus und geben Sie das Ergebnis in lesbarer Form auf der Konsole aus.