

HTW Dresden Fakultät Informatik/Mathematik M. Neugebauer / A. Wülfing	Prüfung I267 – Programmierung von Komponentenarchitekturen	
---	---	--

# Anwendungsfallbeschreibung

## Shop

Für einen Internetshop soll erfasst werden, welche Kunden (Privatkunden, Geschäftskunden) angemeldet sind. Außerdem sollen auch Produkte (Softwaredownload, Hardware) und Bestellungen erfasst werden.

### Beschreibung Datenmodell

Für jeden Kunden soll erfasst werden:

- Adresse (Ort, PLZ, Straße)

Weitere Details müssen für den allgemeinen Kunden nicht erfasst werden. Es soll aber noch in Privatkunden und Geschäftskunden unterschieden werden. Dementsprechend sollen für Privatkunden und Geschäftskunden zusätzlich noch spezifische Eigenschaften erfasst werden. Diese sind für Privatkunden:

- Name
- Vorname

Für Geschäftskunden sollen die folgenden zusätzlichen Eigenschaften erfasst werden:

- Firmenname
- Umsatzsteuer-ID (z. B. DE 123456789)

Außerdem sollen die Produkte für den Internetshop erfasst werden können, wobei jedes Produkt entweder ein Softwareprodukt oder ein Hardwareprodukt ist. Gemeinsam haben beide Produkte eine Produktnummer. Zusätzlich wird für das Softwareprodukt erfasst:

- Produktbezeichnung
- Download-URL
- Speicherplatzbedarf in MB

Für das Hardwareprodukt gibt es die folgenden zusätzlichen Eigenschaften:

- Gewicht
- Produktbezeichnung
- Kurzbeschreibung
- Kategorie der Hardware (Monitor, Rechner, Peripherie, Laptop)

### Sortierung

Für die Hardwareprodukte soll eine Sortierung nach dem Gewicht und der Produktbezeichnung ermöglicht werden.

### Klasse mit verschiedenen Zuständen

In einer Bestellung soll erfasst werden können, welcher Kunde welche Produkte in welcher Stückzahl bestellt. Es soll eine Zeitangabe für die Bestellung und Mengen für die bestellten Produkte in der Bestellung vermerkt werden. Jede Bestellung soll verschiedene Zustände durchlaufen können: BESTELLT, BEARBEITUNG, AUSGELIEFERT.

HTW Dresden Fakultät Informatik/Mathematik M. Neugebauer / A. Wülfing	Prüfung I267 – Programmierung von Komponentenarchitekturen	
---	---	--

## Ereignisse / Zustände

Jeder Wechsel des Zustandes einer Bestellung führt zu einem Ereignis. Zu Beginn befindet sich die Bestellung im Zustand BESTELLT. Wenn das Ereignis „StartBearbeitung“ auftritt, dann wird die Bestellung in den Zustand BEARBEITUNG gesetzt. Wenn danach das Ereignis „BearbeitungFertig“ auftritt, dann wird die Bestellung in den Zustand AUSGELIEFERT überführt. Die Mengen der Produkte in einer Bestellung sollen solange geändert werden können, wie die Bestellung sich im Zustand BEARBEITUNG befindet. Wenn das Ereignis „Mengenaenderung“ auftritt, dann soll eine Mengenänderung in einer existierenden Bestellung passieren.

## Repository

Das Repository soll zusätzlich zu den CRUD-Methoden zurückgeben können, welche Bestellungen im Zustand „BEARBEITUNG“ sind und vor mehr 3 Tagen aufgegeben wurden.

# Aufgaben

1. **Datenmodell:** Entwickeln Sie für das beschriebene Datenmodell geeignete Java-Klassen. Wählen Sie dabei ein geeignetes Vorgehen für Wiederverwendung. Erstellen Sie außerdem ein geeignetes Package, in dem das Datenmodell platziert wird. Eine toString()-Methode kann bei der späteren Ausgabe der Klasse zum einfachen Testen genutzt werden. Implementieren Sie eine geeignete toString()-Methode.
2. **Sortierung:** Für die Sortierung sind zwei Kriterien genannt, nach denen eine Sortierung ermöglicht werden soll. Dafür sind entsprechende Interfaces zu implementieren. Hinweis für String-Attribute: die bei Strings implementierte Methode zum Vergleich darf beim Vergleich der Objekte hinsichtlich des betreffenden Kriteriums eingesetzt werden.
3. **Ereignisse:** Modellieren Sie die Ereignisse, welche im Datenmodell erforderlich sind.
4. **REST:** Erstellen Sie eine REST-Schnittstellen für mindestens eine der vorhandenen Klassen. Die über Beziehungen möglicherweise verbundenen Klassen müssen bei der REST-Schnittstelle nicht einbezogen werden. Es sollen die folgenden Möglichkeiten bestehen:
  - /klassenname - GET-Anfragen aller Objekte, die zu dieser Klasse gehören
  - /klassenname/23 - GET-Anfrage eines Objektes mit einer speziellen ID. Hierbei steht 23 stellvertretend für eine ID und kann auch eine beliebige andere, vorhandene ID sein.
  - /klassenname - POST-Anfrage für das Erstellen eines neuen Objektes der Klasse
  - /klassenname/42 - DELETE-Anfrage für das Löschen des Objektes mit der übergebenen ID. Hierbei steht 42 beispielhaft für die ID des zu löschenden Objektes.
5. **Repository:** Erstellen Sie geeignete Repositories für die implementierten Klassen. Nutzen Sie das CRUD-Repository als Grundlage. Realisieren Sie außerdem die geforderte Abfrage aus der vorherigen Beschreibung des Anwendungsfalles (Abschnitt Repository).
6. **Testen:** Erstellen Sie einen CommandLineRunner, um die Funktionsfähigkeit Ihres Programmcodes zu zeigen. Im CommandLineRunner sollen die folgenden Funktionalitäten Ihrer Implementierung demonstriert werden.
  - Erzeugen Sie konkrete neue Objekte aller Klassen erzeugen und der damit möglicherweise verbundenen Klassen.
  - Speichern Sie die neu erzeugten Objekte in den jeweiligen Repositories.
  - Zeigen Sie die Funktionalität von mindestens zwei verschiedenen Repositories, indem Sie eine Anfrage an das Repository stellen und die Rückgabewerte über ein einfaches System.out oder einen Logger ausgeben.
  - Zeigen Sie die Funktionalität der geforderten zusätzlichen Methode des Repositories (Abschnitt Repository).

HTW Dresden Fakultät Informatik/Mathematik M. Neugebauer / A. Wülfing	Prüfung I267 – Programmierung von Komponentenarchitekturen	
---	---	--

7. **Event-Persistenz:** Konzipieren und implementieren Sie eine Möglichkeit, um die Events dauerhaft zu speichern. Zeigen Sie die Funktionsfähigkeit der Event-Persistenz durch Implementierung in einem weiteren Teil des CommandLineRunners.
8. **Sicherheit:** Perspektivisch soll für den Anwendungsfall der Standard OAuth 2.0 implementiert werden. Dafür sind die Begriffe „Resource Owner“, „Resource Server“, „Authorization Server“ und „Client“ von Bedeutung. Erklären Sie speziell für den vorliegenden Anwendungsfall, auf welchen Teil der Anwendung der jeweilige Begriff zutrifft. Begründen Sie Ihre Zuordnung.
9. **Skalierung:** Das Softwaresystem soll weltweit eingesetzt werden, wodurch wesentlich mehr Nutzer Anfragen stellen und Events erzeugen würden. Beschreiben Sie bezüglich des vorliegenden Anwendungsfalles, wie mit CQRS ein skalierbares System aufgebaut werden kann. Es muss kein Programmcode erstellt werden. Sie sollen nur konzeptionell eine Lösungsmöglichkeit zur Skalierung beschreiben und in einer klar gekennzeichneten Textdatei im Projektordner ablegen.
10. Zusatzaufgabe: Was könnte **Emergenz** in Bezug auf das vorliegende Anwendungsszenario bedeuten?

**Hinweis für Zustände:** Wählen Sie eine geeignete Datenstruktur für das Abbilden der unterschiedlichen Zustände. Die Repräsentation der Zustände durch Zahlen oder Strings ist zulässig, führt aber zu einem geringfügigen Punktabzug. Falls Ihnen keine geeignetere Datenstruktur einfällt, sollten aber trotzdem Zahlen oder Strings verwendet werden, um die weiteren Aufgabenteile lösen zu können.

**Hinweis zu Klassen-, Attribut- und Methodenbezeichnungen:** Prinzipiell ist die gewählte Sprache (englisch oder deutsch) für Ihre Bezeichnungen nicht bedeutsam. Es sollten ein einheitliches Konzept und die geforderten Eigenschaften eindeutig erkennbar sein. Umlaute in Bezeichnungen nicht erlaubt.

**Hinweis zum Vorgehen und zur Bewertung:** Die Aufgabenteile bauen zum Teil aufeinander auf. Für die Bewertung der Teile, welche auf dem Datenmodell aufbauen, ist es zumindest erforderlich ein vereinfachtes Datenmodells zu erstellen. Anhand dieses vereinfachten Datenmodells können Sie dann die darauf aufbauenden Aufgaben lösen und die Funktionsfähigkeit demonstrieren. Für eine positive Bewertung müssen mindestens Ansätze des beschriebenen Datenmodells erkennbar sein. Das bloße Kopieren von Vorlesungs- oder Praktikumsbeispielen führt zu keinen Punkten.