

Bewertungsblatt

In Summe sind 74 Punkte zu erreichen. Für eine Berücksichtigung der Jahresnote müssen mindestens 30 % der Gesamtpunkte (also 23) erreicht werden. Für eine positive Beurteilung der Klausur müssen mindestens 50 % der Gesamtpunkte erreicht werden.

Beurteilungsstufen:

74 – 65 Punkte: Sehr gut, 64 – 56 Punkte: Gut, 55 – 47 Punkte: Befriedigend, 46 – 37 Punkte: Genügend

Aufgabe 1 (30 Punkte in Summe)	Ges	Pkt.
Das erstellte Domainmodel kann vom OR Mapper erzeugt werden.	2	
Das Entity Customer wird korrekt in der erzeugten Datenbank abgebildet.	1	
Die Adresse in Customer ist korrekt als value object konfiguriert.	2	
Das Entity Depot wird korrekt in der erzeugten Datenbank abgebildet.	1	
Das unique constraint in Depot ist korrekt konfiguriert.	1	
Die Adresse in Depot ist korrekt als value object konfiguriert.	2	
Das Entity Driver wird korrekt in der erzeugten Datenbank abgebildet.	1	
Das Feld EmployeeNo ist korrekt als PK definiert.	1	
Das Entity Shipment wird korrekt in der erzeugten Datenbank abgebildet.	1	
Das Feld TrackingNumber ist korrekt als unique definiert.	1	
Die Adresse in Shipment ist korrekt als value object konfiguriert.	2	
Das Feld WeightKg hat den korrekten Datentyp.	1	
Die Enum in Status wird als String in der Datenbank gespeichert.	1	
Das Entity DeliveryAttempt wird korrekt in der erzeugten Datenbank abgebildet.	1	
Der Test PersistDeliveryAttemptTest hat den korrekten Aufbau.	2	
Der Test PersistDeliveryAttemptTest läuft durch.	2	
Der Test EnsureDepotCodeIsUniqueTest hat den korrekten Aufbau.	2	
Der Test EnsureDepotCodeIsUniqueTest läuft durch.	2	
Der Test EnsureTrackingNumberIsUniqueTest hat den korrekten Aufbau.	2	
Der Test EnsureTrackingNumberIsUniqueTest läuft durch.	2	

Aufgabe 2 (16 Punkte in Summe)	Ges	Pkt.
Die Methode GetCategoriesWithProductCounts ist korrekt.	2	
Die Methode GetPreordersOfCustomer ist korrekt.	3	
Die Methode GetRevenueOfProduct ist korrekt.	3	
Die Methode AddPreorder berücksichtigt korrekt eine ungültige CustomerId.	1	

Aufgabe 2 (16 Punkte in Summe)	Ges	Pkt.
Die Methode AddPreorder berücksichtigt korrekt eine leere Liste productPreorders.	2	
Die Methode AddPreorder fügt korrekt den Datensatz in Preorder ein.	2	
Die Methode AddPreorder fügt korrekt die Datensätze in PreorderItem ein.	3	

Aufgabe 3 (28 Punkte in Summe)	Ges	Pkt.
GET /customers liefert HTTP 200.	1	
GET /customers liefert die korrekten Daten.	2	
GET /customers/preorder/{code} liefert HTTP 200 bei gültigem Code.	1	
GET /customers/preorder/{code} liefert HTTP 400 bei zu kurzem Code.	1	
GET /customers/preorder/{code} liefert HTTP 404 bei nicht vorhandenem Code.	1	
GET /customers/preorder/{code} liefert die korrekten Daten.	2	
POST /customers liefert HTTP 400 bei ungültigem Firstname.	1	
POST /customers liefert HTTP 400 bei ungültigem Lastname.	1	
POST /customers liefert HTTP 400 bei ungültiger Email.	1	
POST /customers liefert HTTP 201 bei gültigen Daten.	1	
POST /customers fügt den Datensatz in Customer ein.	2	
DELETE /customer/{id} liefert HTTP 404, wenn die CustomerId nicht existiert.	1	
DELETE /customer/{id} liefert HTTP 400, wenn der Kunde Vorbestellungen hat.	1	
DELETE /customer/{id} liefert HTTP 204, wenn der Kunde gelöscht werden konnte.	1	
DELETE /customer/{id} löscht den Datensatz in Customer.	2	
Der Integration Test für den DELETE Endpunkt prüft korrekt das Verhalten bei ungültiger CustomerId.	3	
Der Integration Test für den DELETE Endpunkt prüft korrekt das Verhalten bei Kunden mit Vorbestellungen.	3	
Der Integration Test für den DELETE Endpunkt prüft korrekt das Verhalten beim erfolgreichen Löschen des Kunden.	3	