

IHK Anwendungsentwickler Prüfung 2025 Praktisch – Lösungen

Teil A – Anforderungsanalyse und Systementwurf (20 Punkte)

Aufgabe A1 (7 Punkte)

- a) Ein Entity-Relationship-Diagramm sollte mindestens die Entitäten Kunde, Veranstaltung, Buchung, Sitzplatz und Zahlung enthalten. Kardinalitäten könnten wie folgt aussehen: Ein Kunde kann mehrere Buchungen haben (1:n), eine Buchung gehört zu genau einer Veranstaltung (n:1), eine Buchung kann mehrere Sitzplätze umfassen (1:n), und eine Buchung kann mit einer Zahlung verknüpft sein (1:1).
- b) In der Entität „Buchung“ sollten Attribute wie BuchungsID, KundelD, VeranstaltungsID, Datum und Status enthalten sein. In der Entität „Zahlung“ sollten Attribute wie ZahlungsID, BuchungsID, Betrag, Zahlungsdatum und Zahlungsart enthalten sein.

Aufgabe A2 (7 Punkte)

Ein UML-Klassendiagramm sollte die Klassen Kunde, Buchung, Veranstaltung und Zahlung mit mindestens drei Attributen pro Klasse enthalten. Beispiel: Kunde (KundenID: int, Name: String, Email: String), Buchung (BuchungsID: int, KundelD: int, VeranstaltungsID: int), Veranstaltung (VeranstaltungsID: int, Name: String, Datum: Date), Zahlung (ZahlungsID: int, BuchungsID: int, Betrag: double). Zwei Assoziationen könnten zwischen Kunde und Buchung (1:n) sowie Buchung und Zahlung (1:1) bestehen. Jede Klasse sollte eine Methode enthalten, z.B. Kunde: registrieren().

Aufgabe A3 (6 Punkte)

Use-Case „Ticket buchen“: Akteur: Kunde, Vorbedingung: Kunde ist eingeloggt, Ablauf: 1. Veranstaltung auswählen, 2. Sitzplatz wählen, 3. Buchung bestätigen, 4. Zahlung durchführen, Nachbedingung: Buchung erfolgreich. Use-Case „Buchung stornieren“: Akteur: Kunde, Vorbedingung: Buchung existiert, Ablauf: 1. Buchung auswählen, 2. Stornierung bestätigen, 3. Rückerstattung veranlassen, Nachbedingung: Buchung storniert. Use-Case „Zahlung verarbeiten“: Akteur: System, Vorbedingung: Buchung abgeschlossen, Ablauf: 1. Zahlungsdaten abrufen, 2. Zahlung validieren, 3. Zahlung ausführen, Nachbedingung: Zahlung erfolgreich.

Teil B – Implementierung und Testing (30 Punkte)

Aufgabe B1 (10 Punkte)

- a) Der Fehler liegt darin, dass die Liste „items“ null sein könnte. Behebung: Überprüfen Sie, ob „items“ null ist, bevor Sie darauf zugreifen.
- b) Änderung: Fügen Sie eine Null-Prüfung hinzu, z.B. if (items == null) return 0;.
- c) Defensive-Programming-Maßnahmen: 1. Überprüfen Sie, ob jedes „OrderItem“ in der Liste null ist, bevor Sie darauf zugreifen. 2. Verwenden Sie Optional für Rückgabewerte, um NullPointerExceptions zu vermeiden.

Aufgabe B2 (8 Punkte)

- a) Die Laufzeitkomplexität der rekursiven Fibonacci-Methode ist $O(2^n)$.
- b) Optimierter Algorithmus (Pseudocode):

```
...  
function fib(n):  
    if n <= 1: return n  
    fibs = array of size n+1  
    fibs[0] = 0  
    fibs[1] = 1  
    for i from 2 to n:  
        fibs[i] = fibs[i-1] + fibs[i-2]  
    return fibs[n]  
...
```

- c) Vorteil: Die optimierte Methode hat eine Laufzeitkomplexität von $O(n)$ und verwendet eine iterative Lösung, die den Speicherverbrauch reduziert.

Aufgabe B3 (7 Punkte)

JUnit-Testfälle:

- Test mit net = 100, vatRate = 0.2, erwartet: 120.
- Test mit net = 0, vatRate = 0.2, erwartet: 0.
- Test mit net = 100, vatRate = 0, erwartet: 100.
- Test mit net = -100, vatRate = 0.2, erwartet: IllegalArgumentException.
- Test mit net = 100, vatRate = -0.2, erwartet: IllegalArgumentException.

Aufgabe B4 (5 Punkte)

Maßnahmen zur Performancesteigerung:

- Caching: Verwenden Sie Caching für häufig abgerufene Daten, um die Datenbanklast zu reduzieren.
- Asynchrone Verarbeitung: Verwenden Sie asynchrone Aufrufe, um die Reaktionszeit zu verbessern.
- Optimierung der Datenbankabfragen: Verwenden Sie Indizes und optimieren Sie SQL-Abfragen, um die Abfragegeschwindigkeit zu erhöhen.

Teil C – Datenbanken und Schnittstellen (20 Punkte)

Aufgabe C1 (10 Punkte)

a) SQL-Abfrage:

```
...  
SELECT Kunden.Name, COUNT(Bestellungen.BestellID) AS AnzahlBestellungen, SUM(Bestellungen.Gesamtbetrag) AS SummeBetrag  
FROM Kunden  
JOIN Bestellungen ON Kunden.KundenID = Bestellungen.KundenID  
GROUP BY Kunden.Name;  
...
```

- b) Indizes: Ein Index auf „KundenID“ in beiden Tabellen würde die JOIN-Operation beschleunigen. Ein zusätzlicher Index auf „BestellID“ in der Tabelle „Bestellungen“ könnte die Abfrage weiter optimieren.

Aufgabe C2 (6 Punkte)

REST-API-Endpoint-Design:

- URL: /api/bestellungen, HTTP-Methode: POST
- JSON-Schema für Request: { "kundenID": int, "datum": string, "gesamtbetrag": double }
- JSON-Schema für Response: { "bestellID": int, "status": string }
- HTTP-Statuscodes: 201 Created (Erfolg), 400 Bad Request (ungültige Daten), 500 Internal Server Error (Serverfehler)

Aufgabe C3 (4 Punkte)

Migrationsplan:

- Erstellen Sie ein Backup der aktuellen Datenbank.
- Fügen Sie das neue Feld „Status“ zur Tabelle „Bestellungen“ hinzu mit Default-Wert „PENDING“.
- Testen Sie die Änderungen in einer Staging-Umgebung.
- Führen Sie die Migration in der Produktionsumgebung durch.
- Überwachen Sie das System auf Fehler nach der Migration.

Teil D – Betrieb, Sicherheit und Wartung (15 Punkte)

Aufgabe D1 (6 Punkte)

Sicherheitsanforderungen:

- Eingabevalidierung (A01: Injection)
- Authentifizierung und Autorisierung (A07: Identification and Authentication Failures)
- Verschlüsselung sensibler Daten (A02: Cryptographic Failures)
- Sicherheitskonfigurationen (A05: Security Misconfiguration)
- Schutz vor Cross-Site Scripting (A03: Injection)

Aufgabe D2 (5 Punkte)

Backup-Konzept:

- Backup-Typen: Voll-Backup wöchentlich, inkrementelle Backups täglich.
- Aufbewahrungsfristen: Voll-Backups 1 Monat, inkrementelle Backups 1 Woche.
- Medien/Technologien: Cloud-Speicher für schnelle Wiederherstellung, externe Festplatten für Langzeitarchivierung.

Aufgabe D3 (4 Punkte)

Logging und Monitoring:

- Logging-Framework: Log4j, Log-Level: INFO, ERROR
- Metriken: Antwortzeit der API, Anzahl der Fehlermeldungen
- Alarmbedingungen: Antwortzeit > 2 Sekunden, mehr als 5 Fehler in 10 Minuten

Teil E – Projektmanagement und Dokumentation (15 Punkte)

Aufgabe E1 (9 Punkte)

Projektplan (ASCII-Gantt-Diagramm):

```
...  
Anforderungsanalyse: |#####|  
Design:              |#####|  
Implementierung:    |#####|  
Test:               |#####|  
Rollout:            |###|  
...  
Meilensteine:  
1. Abschluss der Anforderungsanalyse (Monat 1)  
2. Fertigstellung des Designs (Monat 2)  
3. Abschluss der Implementierung (Monat 4)  
4. Abschluss der Tests (Monat 5)  
5. System-Rollout (Monat 6)
```

Aufgabe E2 (6 Punkte)

Gliederung der technischen Dokumentation:

- Einleitung: Überblick über das System und seine Ziele.
- Systemarchitektur: Beschreibung der Systemkomponenten und ihrer Interaktionen.
- Installationsanleitung: Schritt-für-Schritt-Anleitung zur Installation des Systems.
- Benutzerhandbuch: Anweisungen zur Nutzung des Systems für Endbenutzer.
- Administratorhandbuch: Informationen zur Systemkonfiguration und -wartung.
- Fehlerbehebung: Häufige Probleme und deren Lösungen.