

IHK Anwendungsentwickler Prüfung 2024 Praktisch – Lösungen

Teil 1: Programmierung

1.1 Methode analysieren

a) Ablauf:

- Initialisierung einer Datenstruktur zur Speicherung eindeutiger Elemente.
- Iteration über das String-Array.
- Überprüfung, ob das Element bereits in der Datenstruktur vorhanden ist.
- Falls nicht vorhanden, Hinzufügen des Elements.
- Sortierung der gespeicherten Elemente.
- Rückgabe der sortierten, eindeutigen Elemente.

b) Laufzeitkomplexitäten:

- Best Case: $O(n \log n)$ (wenn alle Elemente bereits eindeutig sind und nur sortiert werden müssen).
- Worst Case: $O(n^2)$ (wenn alle Elemente Duplikate sind und die Überprüfung auf Eindeutigkeit teuer ist).

1.2 UML-Klassendiagramm

- Klassen: Kunde, Bestellung, Artikel.
- Attribute: Kunde (KundenID, Name), Bestellung (BestellID, Datum), Artikel (ArtikelID, Bezeichnung, Preis).
- Methoden: Kunde (bestellen()), Bestellung (hinzufügenArtikel()), Artikel (getPreis()).
- Beziehungen: Kunde 1..* Bestellung, Bestellung 1..* Artikel.
- Sichtbarkeiten: private für Attribute, public für Methoden.
- Kardinalitäten: Kunde 1..* Bestellung, Bestellung 1..* Artikel.

1.3 Refactoring

- Verstoß 1: Datenbankzugriff in der Geschäftslogik. Maßnahme: Auslagern in eine separate Datenzugriffsklasse.
- Verstoß 2: UI-Logik in der Geschäftslogik. Maßnahme: Trennung in eine separate UI-Klasse.
- Verstoß 3: Validierungslogik in der Geschäftslogik. Maßnahme: Auslagern in eine Validierungsklasse.

1.4 Fehleranalyse

- Ursache 1: Unzureichende Sperrmechanismen. Lösung: Implementierung von Locking-Mechanismen.
- Ursache 2: Falsche Reihenfolge beim Sperren von Ressourcen. Lösung: Konsistente Sperrreihenfolge einführen.

Teil 2: Datenbankentwicklung

2.1 ER-Modell

a) Entitäten: Fahrzeug, Fahrer, Einsatz.

- Schlüsselattribute: Fahrzeug (FahrzeugID), Fahrer (FahrerID), Einsatz (EinsatzID).
- Beziehungstypen: Fahrzeug 1..1 Fahrer, Fahrer 0..* Fahrzeug, Einsatz 1..1 Fahrzeug, Einsatz 1..1 Fahrer.

b) Kardinalitäten: Ein Fahrzeug kann nur von einem Fahrer gleichzeitig genutzt werden, ein Fahrer kann mehrere Fahrzeuge fahren.

2.2 SQL-Abfragen

- a) `SELECT Abteilung.Bezeichnung FROM Abteilung JOIN Mitarbeiter ON Abteilung.ID = Mitarbeiter.AbteilungID GROUP BY Abteilung.Bezeichnung HAVING AVG(Mitarbeiter.Gehalt) > 50000;`
- b) `SELECT Abteilung.Bezeichnung, COUNT(Mitarbeiter.ID) FROM Abteilung JOIN Mitarbeiter ON Abteilung.ID = Mitarbeiter.AbteilungID GROUP BY Abteilung.Bezeichnung HAVING AVG(Mitarbeiter.Gehalt) > 50000;`

2.3 Normalisierung

- 1NF: Verstoß durch nicht atomare Werte (z.B. Kundenname, Kundenadresse). Zerlegung: Trennung in separate Tabellen für Kunden.
- 2NF: Verstoß durch funktionale Abhängigkeiten (z.B. Artikelbezeichnung von ArtikelID). Zerlegung: Trennung in separate Tabellen für Artikel.
- 3NF: Verstoß durch transitive Abhängigkeiten (z.B. Kundenadresse von Kundenname). Zerlegung: Trennung in separate Tabellen für Kundeninformationen.

Teil 3: Software-Architektur & Entwurf

3.1 Design-Pattern-Auswahl

a) Singleton-Pattern.

- b) Klassendiagramm: Eine Klasse mit privatem Konstruktor, statischer Methode `getInstance()`, statisches Attribut `instance`.

3.2 Schichtenmodell erklären

- a) Präsentation: Darstellung der Daten für den Benutzer.
- Geschäftslogik: Verarbeitung der Daten und Geschäftsregeln.
- Datenzugriff: Zugriff auf die Datenbank.
- b) Vorteile: Bessere Wartbarkeit, Wiederverwendbarkeit der Schichten.
- c) Gefahr: Enge Kopplung bei unsauberer Trennung.

3.3 Deploymentdiagramm

- Knoten: Load Balancer, App-Server (3x), Datenbank, Monitoring-Agent.
- Verbindungen: Load Balancer zu App-Servern, App-Server zu Datenbank.
- Ports: HTTP/HTTPS für Load Balancer, spezifische Ports für Datenbankzugriff.

Teil 4: IT-Projektmanagement & Kalkulation

4.1 Aufwandsschätzung

- Formel: $PM = 2,94 * 1,2 * (1200^{1,1})$.
- Ergebnis: $PM \approx 374$ Person-Months.

4.2 Netzplan

a) Aufgaben: A, B, C, D, E.

- Abhängigkeiten: C nach A, D nach A, E nach B und C.
- b) Kritischer Pfad: A -> B -> E.

4.3 Kostenberechnung

- Stundenpreis: $80 \text{ EUR} * 1,3 * 1,1 = 114,4 \text{ EUR}$.
- Gesamtprojektpreis: $114,4 \text{ EUR} * 150 \text{ Std.} = 17.160 \text{ EUR}$.

Teil 5: Fachenglisch

5.1 Begriffsübersetzung

a) Datensicherung: Data backup

- Lastverteilung: Load balancing
- Ausfallsicherheit: Fault tolerance
- Quellcodeverwaltung: Source code management
- Entwicklungsumgebung: Development environment

5.2 Textzusammenfassung

- Microservices sind kleine, unabhängige Dienste, die zusammenarbeiten, um eine Anwendung zu bilden, was Flexibilität und Skalierbarkeit erhöht.

5.3 E-Mail

Dear [Customer Name],

Thank you for reaching out. We are currently working on the bug fix and expect to have it resolved by [Date]. We appreciate your patience and will keep you updated on any changes.

Best regards,
[Your Name]