

Mündlichen Abiturprüfung – Informatik

Thema: OOP, Algorithmen, Daten Banken, Formale Sprachen

30. Mai 2025

Aufgabe 1: Analyse eines Algorithmus

Ein unbekannter Algorithmus zur Berechnung einer speziellen mathematischen Funktion von Zahlen wird bereitgestellt. Der Name der Klasse lautet `Algorithmus`.

Gegebener Java-Code

```
public class Algorithmus {

    public static int berechne(int b, int e) {
        if (e == 0) {
            return 1;
        }
        return b * berechne(b, e - 1);
    }

    public static void main(String[] args) {
        int b = 3;
        int e = 3;
        System.out.println("Ergebnis: " + berechne(b, e));
    }
}
```

Teilaufgaben

1. Beschreiben Sie die Funktionsweise des Algorithmus. Welche mathematische Berechnung führt die Methode `berechne` der Klasse `Algorithmus` durch?
2. Welche Art der Implementierung wurde gewählt und welche Vor- oder Nachteile ergeben sich daraus?
3. Bestimmen Sie die Zeitkomplexität des Algorithmus.
4. Vergleichen Sie die rekursive Lösung mit einer iterativen Variante hinsichtlich Aufwand, Verständlichkeit und praktischer Anwendung.

Aufgabe 2: Datenmodellierung und Normalisierung – Kontext: Arztpraxis

Eine Arztpraxis verwaltet Informationen über Patienten, behandelnde Ärzte und durchgeführte Behandlungen in einer relationalen Datenbank. Die ursprüngliche Tabellenstruktur ist wie folgt aufgebaut:

Tabelle 1: Ursprüngliche nicht normalisierte Tabelle

PatientID	Name	Adresse (Str., Stadt, PLZ)	Geburtsdatum	ArztID	Arztname	Fachgebiet	Diagnose
P01	Anna Müller	Blumenweg 12, Köln, 50667	12.03.1985	A01	Dr. Schmitt	Kardiologie	Bluthochdruck
P02	Bernd Meyer	Marktstr. 5, Bonn, 53111	04.09.1978	A02	Dr. Klein	Orthopädie	Bandscheibenvorfall
P01	Anna Müller	Blumenweg 12, Köln, 50667	12.03.1985	A02	Dr. Klein	Orthopädie	Rückenschmerzen
P03	Carla Schulz	Parkallee 7, Köln, 50668	22.06.1992	A01	Dr. Schmitt	Kardiologie	Herzrhythmusstörung

Teilaufgaben

1. Identifizieren Sie Redundanzen und Anomalien in der Tabelle. (10 BE)
2. Zerlegen Sie die nicht normalisierte Tabelle schrittweise und normalisieren Sie sie bis zur 3. Normalform. Geben Sie die neu entstandenen Tabellen an (ohne Daten). (20 BE)
3. Erläutern Sie, welche konkreten Vorteile die Normalisierung für die Verwaltung einer Praxisdatenbank bringt. (10 BE)

Aufgabe 3: Analyse einer formalen Grammatik zur Passworterstellung

Gegeben sei folgende formale Grammatik G bestehend aus Nichtterminalen, Terminalen, Produktionsregeln und einem Startsymbol:

```

<passwort> → <zeichen><zeichen><zeichen><zeichen><zeichen>
<zeichen> → <klein> | <gross> | <ziffer> | <sonder>
<klein> → a, b, c, ...
<gross> → A, B, C, ...
<ziffer> → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
<sonder> → !, ?, §, *

```

Teilaufgaben:

1. Leiten Sie das Passwort aA1! mit der Grammatik ab.
2. Gehört das Passwort Pass! zur Sprache $L(G)$? Begründen Sie Ihre Antwort.
3. Die Erweiterung der Grammatik, um sichere Passwörter zu erzeugen (z. B. mit variabler Länge, echten Buchstabenklassen oder strukturellen Sicherheitsregeln), ist formal komplex.

Diskutieren Sie, in welchen Situationen es dennoch sinnvoll sein kann, solche Grammatiken oder regelbasierten Modelle in sicherheitskritischer Software einzusetzen – etwa bei der automatisierten Prüfung oder Generierung von Passwörtern.

Gehen Sie auch auf moderne Authentifizierungsverfahren ein, bei denen anstelle klassischer Passwörter biometrische Merkmale wie Gesichtserkennung oder Fingerabdruck verwendet werden.

Lösungen

Lösung zu Aufgabe 2: Datenmodellierung und Normalisierung – Praxisdatenbank

1. Redundanzen und Anomalien:

- **Redundanz:**

- Patientendaten (z. B. Name, Adresse, Geburtsdatum) erscheinen mehrfach (z. B. PatientID P01).
- Arztinformationen (z. B. Name, Fachgebiet) sind für dieselbe ArztID mehrfach vorhanden.

- **Anomalien:**

- **Änderungsanomalie:** Ändert sich die Telefonnummer oder Adresse eines Arztes oder Patienten, müssen mehrere Zeilen angepasst werden.
- **Einfügeanomalie:** Ein neuer Arzt oder Patient kann nicht ohne vollständigen Behandlungseintrag erfasst werden.
- **Löschanomalie:** Löscht man eine Behandlung, könnten wichtige Informationen über Patienten oder Ärzte verloren gehen.

2. Schrittweise Normalisierung bis zur 3. Normalform:

1NF – Atomare Werte sicherstellen:

Die Spalte „Adresse“ enthält zusammengesetzte Werte. → Aufspaltung in Straße, Stadt, PLZ.

Ergebnis: Alle Attributwerte sind atomar.

2NF – Beseitigung partieller Abhängigkeiten:

Die Kombination PatientID + ArztID ist zusammengesetzter Primärschlüssel der Ausgangstabelle. Einige Attribute hängen nur von einem Teil des Schlüssels ab (z. B. Name des Patienten nur von PatientID).

Zerlegung:

- Patient(PatientID, Name, AdresseID, Geburtsdatum)
- Arzt(ArztID, Name, Fachgebiet)
- Adresse(AdresseID, Straße, Stadt, PLZ)

3NF – Beseitigung transitiver Abhängigkeiten:

Alle Nichtschlüsselattribute hängen direkt und ausschließlich vom Primärschlüssel ihrer Tabelle ab. → Keine weitere Zerlegung notwendig.

Zusätzlich:

- Behandlung(PatientID, ArztID, Diagnose)

Endgültige Tabellenstruktur mit Schlüsseln:

- Patient(PatientID, Name, AdresseID (FK), Geburtsdatum)
 - Arzt(ArztID, Name, Fachgebiet)
 - Adresse(AdresseID, Straße, Stadt, PLZ)
 - Behandlung(PatientID (FK), ArztID (FK), Diagnose)
-
- **Vermeidung von Redundanzen:** Informationen zu Patienten und Ärzten werden nur einmal gespeichert.
 - **Einfache Pflege:** Änderungen (z. B. Adresswechsel) müssen nur an einer Stelle vorgenommen werden.
 - **Datenkonsistenz:** Keine widersprüchlichen Angaben möglich.
 - **Erweiterbarkeit:** Neue Patienten, Ärzte oder Diagnosen können unabhängig voneinander hinzugefügt werden.
 - **Datenintegrität:** Fremdschlüsselbeziehungen sichern die logische Verknüpfung der Daten.