
Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
---------------------------	-----------------------	-----------------------------

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2021**

66116

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
— Prüfungsaufgaben —**

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbanksysteme, Softwaretechnologie**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **24**

Bitte wenden!

Thema Nr. 1**Teilaufgabe I: Softwaretechnologie****Aufgabe 1****(6 Punkte)**

- a) Definieren Sie die Bedeutung der Sichtbarkeiten private, package-private und protected von Feldern in Java-Klassen. Erklären Sie diese kurz (je ein Satz).
- b) Benennen Sie jeweils einen Grund für den Einsatz der Sichtbarkeiten private, package-private und protected von Feldern in Java-Klassen.

Aufgabe 2**(12 Punkte)**

Lesen Sie die folgenden Beispielcodes gründlich. Identifizieren Sie für jeden Beispielcode den jeweiligen wesentlichen Verstoß gegen die Prinzipien guter objektorientierter Programmierung. Benennen und erklären Sie jeweils den Verstoß (Fehler) in einem Satz und erläutern Sie für jeden Beispielcode, welche Probleme aus dem jeweiligen Fehler resultieren können, ebenfalls in einem Satz.

Beispielcode 1

```
class Rectangle {
    private int width;
    private int length;

    Rectangle(int w, int l) {
        width = w;
        length = l;
    }

    public int getWidth() {
        return this.width;
    }

    public int getLength() {
        return this.length;
    }
}

class RectangleDemo {
    public static void main (String args[]) {
        Rectangle rectangle1 = new Rectangle(10, 20);
        Rectangle rectangle2 = new Rectangle(3, 90);
        Rectangle example = new Rectangle(1, 2);
        int area;

        // Compute area of first box
        area = rectangle1.getWidth() * rectangle1.getLength();
        System.out.println("Area is " + area);
    }
}
```

Fortsetzung nächste Seite!

```
// Compute area for second box
area = rectangle2.getWidth() * rectangle2.getLength();
System.out.println("Area is" + area);

// Compute area for third box
area = example.getWidth() * example.getLength();
System.out.println("Area is" + area);
}
}
```

Beispielcode 2

```
public class CalculateSpeed {
    private double kilometers;
    private double minutes;

    public CalculateSpeed (double k, double m) {
        this.kilometers = k;
        this.minutes = m;
    }

    // Display the speed
    void speed() {
        double speed;
        speed = kilometers / (minutes / 60);
        System.out.println("A car traveling " + kilometers +
            " kilometers in " + minutes +
            " minutes travels at " + speed +
            " kilometers per hour");
    }

    public static void main(String args[]) {
        CalculateSpeed car = new CalculateSpeed(110.0, 120.0);

        // Display car speed
        car.speed();

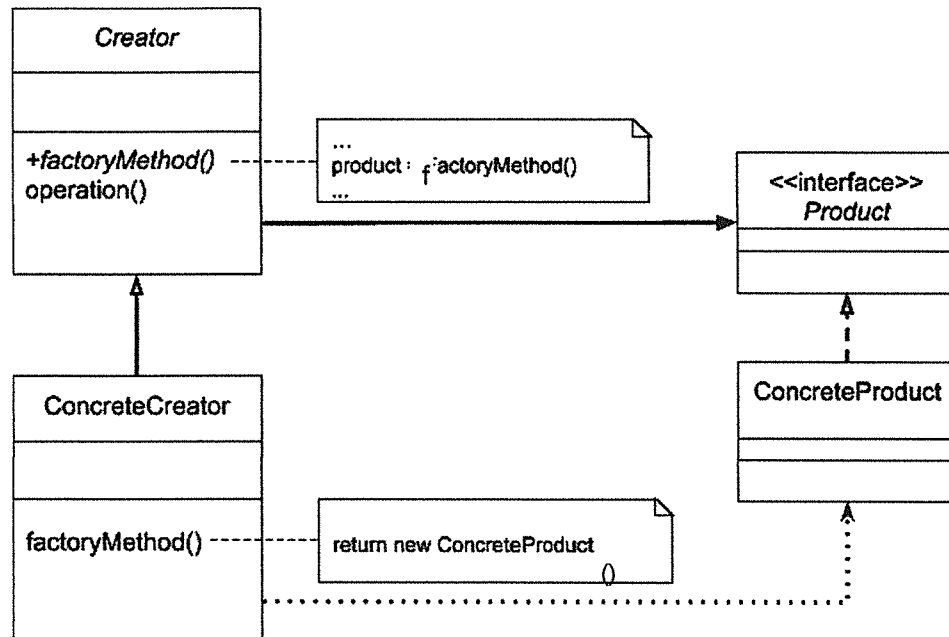
        // Display bicycle speed
        double speed;
        speed = 20.0 / (80.0 / 60);
        System.out.println("A bicycle traveling " + kilometers +
            " kilometers in " + minutes +
            " minutes travels at " + speed +
            " kilometers per hour");
    }
}
```

Beispielcode 3

```
class Stack {  
    int stck[] = new int[3];  
    public int top;  
  
    // Initialize top of stack  
    Stack() {  
        top = -1;  
    }  
  
    // Push an item on the stack  
    void push(int item) {  
        if (top==2){  
            System.out.println("Stack is full.");  
        } else {  
            stck[++top] = item;  
        }  
    }  
  
    // Pop an item from the stack  
    int pop() {  
        if (top < 0) {  
            throw new IllegalStateException("Stack is empty.");  
        } else {  
            return stck[top--];  
        }  
    }  
}
```

Aufgabe 3**(12 Punkte)**

Wählen Sie bis zu fünf unterschiedliche Elementtypen aus folgendem Diagramm aus und benennen Sie diese Elemente und ihre **syntaktische** (nicht semantische) Bedeutung.

**Aufgabe 4****(12 Punkte)**

- a) Schreiben Sie ein Programm in einer objektorientierten Programmiersprache Ihrer Wahl, das den folgenden Anweisungen entspricht.
1. Es gibt eine Klasse mit dem Namen `Box`.
 2. Alle Zahlen sind Fließkommazahlen.
 3. `Box` wird mit einem Argument instanziiert, dessen Wert einer Variable namens `length` zugewiesen wird.
 4. `Box` hat eine Methode ohne Argumente namens `size`, welche den Wert von `length` zurückgibt.
 5. Eine weitere Methode namens `size` hat genau ein Argument namens `width`. Diese zweite Methode namens `size` gibt das Produkt aus `width` und `length` zurück.
 6. Eine weitere Methode namens `size` hat genau zwei Argumente, nämlich eine Zahl `num` und einen Faktor `f`. Es wird `length` minus das Produkt aus `num` und `f` zurückgegeben.
 7. Schreiben Sie eine `main` Methode, die eine `Box` namens `example` mit einer Länge von 15 anlegt.
 8. Führen Sie die Methode `size` in der `main` Methode wie unten angegeben drei Mal aus. Speichern Sie hierbei das Ergebnis jeweils in einer Variable `mysize`. Geben Sie das Ergebnis jeweils in einer eigenen Zeile des Ausgabemediums `System.out` aus.
 - Mit keinen Argumenten
 - Mit dem Argument 10
 - Mit den beiden Argumenten 5 und 2

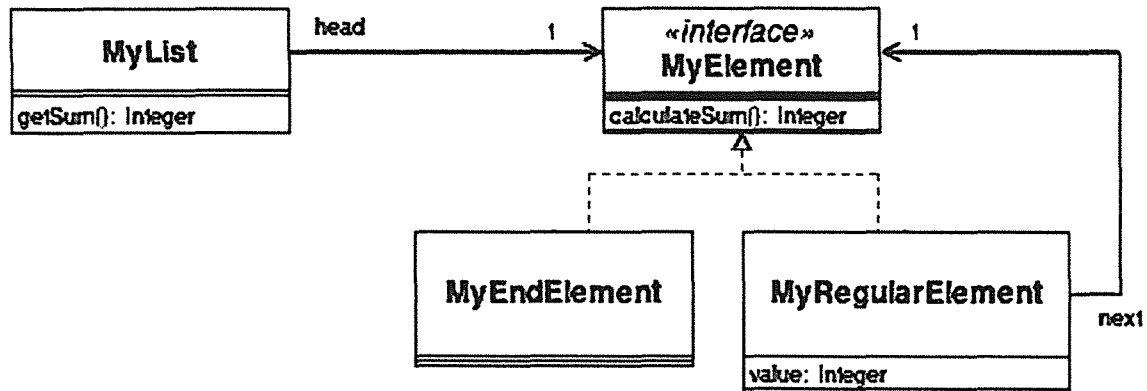
Fortsetzung nächste Seite!

b) Notieren Sie die Ausgabe der main-Methode.

Aufgabe 5

(16 Punkte)

Die folgende Abbildung stellt den Entwurf der Implementierung einer verketteten Liste dar, welche Integer-Werte als Elemente enthalten kann.



Die Klasse MyList stellt die Methode getSum() zur Verfügung, welche die Summe über alle in einer Liste befindlichen Elemente berechnet. Ein Ausschnitt der Implementierung sieht folgendermaßen aus:

```

public class MyList {
    private MyElement head;
    ...
    public MyList() {
        this.head = new MyEndElement();
    }
    ...
    public int getSum() {
        ...
    }
}
  
```

Gehen Sie im Folgenden davon aus, dass bereits Methoden existieren, welche Elemente in die Liste einfügen können.

Fortsetzung nächste Seite!

- a) Implementieren Sie in einer objektorientierten Programmiersprache Ihrer Wahl, z. B. Java, die Methode `calculateSum()` der Klassen `MyEndElement` und `MyRegularElement`, so dass rekursiv die Summe der Elemente der Liste berechnet wird. Als Abbruchbedingung darf hierbei nicht das Feld `MyRegularElement.next` auf den Wert "null" überprüft werden.

Hinweis: Gehen Sie davon aus, die Implementierung von `MyList` garantiert, dass `MyRegularElement.next` niemals den Wert "null" annimmt, sondern das letzte hinzugefügte `MyRegularElement` auf eine Instanz der Klasse `MyEndElement` verweist. Es gibt immer nur eine Instanz der Klasse `MyEndElement` in einer Liste.

Hinweis: Achten Sie auf die Angabe einer korrekten Methodensignatur.

- b) Nennen Sie den Namen des Entwurfsmusters, auf welchem das oben gegebene Klassendiagramm basiert, und ordnen Sie dieses in eine der Kategorien von Entwurfsmustern ein.

Hinweis: Es genügt die Angabe eines Musters, falls Sie mehrere Muster identifizieren sollten.

Aufgabe 6

(21 Punkte)

- a) Erläutern Sie den Zweck (Intent) des Erzeugungsmusters Erbauer in max. drei Sätzen, ohne dabei auf die technische Umsetzung einzugehen.
- b) Erklären Sie, wie das Erzeugungsmuster Erbauer umgesetzt werden kann (Implementierung). Die Angabe von Code ist hierbei NICHT notwendig!
- c) Nennen Sie jeweils einen Vor- und einen Nachteil des Erzeugungsmusters Erbauer im Vergleich zu einer Implementierung ohne dieses Muster.

Fortsetzung nächste Seite!

Aufgabe 7**(5 Punkte)**

Lesen Sie die folgenden alternativen Codestücke.

```
public class MyParser {  
    private InputStream input;  
    ...  
    public MyParser(String filePath) {  
        ...  
    }  
}  
  
public class MyParser {  
    private InputStream input;  
    ...  
    public MyParser(InputStream stream) {  
        ...  
    }  
}
```

Beide Codestücke zeigen die Initialisierung einer Klasse namens MyParser. Das zweite Codestück nutzt jedoch hierfür eine Technik namens Abhängigkeits-Injektion (Dependency Injection).

- a) Erklären Sie den Unterschied zwischen beiden Initialisierungen.
Hinweis: Sie können diese Aufgabe auch lösen, falls Sie die Technik nicht kennen.
- b) Benennen Sie einen Vorteil dieser Technik.

Aufgabe 8**(3 Punkte)**

Das Client-Server-Modell ist ein Architekturmuster. Nennen Sie zwei Vorteile einer nach diesem Muster gestalteten Architektur.

Aufgabe 9**(8 Punkte)**

Betrachten Sie die folgende Liste von Technologien:

- A Node.js
- B PHP
- C CSS
- D AJAX
- E Python
- F Java

Welche dieser Technologien laufen in einem Client-Server-System üblicherweise auf der Seite des Klienten und welche auf der Seite des Servers? Nehmen Sie hierzu an, dass der Client ein Browser ist.

Übertragen Sie die im Folgenden gegebene Tabelle in Ihren Bearbeitungsbogen und ordnen Sie die aufgelisteten Technologien anhand der Buchstaben in die Tabelle ein. **Fortsetzung nächste Seite!**

Hinweis: Mehrfachzuordnungen sind möglich.

Client-seitige Technologien	Server-seitige Technologien

Aufgabe 10

(4 Punkte)

- Was bedeutet die Abkürzung AJAX?
- Erklären Sie in max. drei Sätzen die grundlegende Funktion von AJAX.

Aufgabe 11

(8 Punkte)

- Was ist das Hypertext Transfer Protocol (HTTP) und wozu dient es?
- Betrachten Sie die folgende Zeile Text. Um welche Art von Text handelt es sich?
<https://developer.mozilla.org/en-US/search?q=client+server+overview>
- Was sind die vier wesentlichen Bestandteile des Texts aus der vorigen Teilaufgabe?

Aufgabe 12

(13 Punkte)

Es gibt Softwaresysteme, welche auf peer-to-peer (P2P) Kommunikation basieren und eine entsprechende Architektur aufweisen.

- Bewerten Sie die folgenden Aussagen als entweder richtig oder falsch. Notieren Sie zu jeder Nummer Ihre Entscheidung in Ihren Bearbeitungsbogen. (Keine Begründung erforderlich.)

Nr.	Richtig?	Falsch?	Aussage
1			Mithilfe des Befehls "lookup" können Peers sich gegenseitig identifizieren.
2			In einem P2P-System, wie auch bei Client-Server, sind alle Netzwerkteilnehmer gleichberechtigt.
3			Alle P2P-Systeme funktionieren grundsätzlich ohne einen zentralen Verwaltungs-Peer.
4			P2P kann auch für eine Rechner-Rechner-Verbindung stehen.
5			Es gibt strukturierte und unstrukturierte P2P-Systeme. In unstrukturierten P2P-Systemen wird zum Auffinden von Peers eine verteilte Hashtabelle verwendet (DHT).
6			In einem P2P-System sind theoretisch alle Peers gleichberechtigt, praktisch gibt es jedoch leistungsabhängige Gruppierungen.
7			Ein Peer kann sowohl ein Client wie auch ein Server für einen anderen Peer sein.

Fortsetzung nächste Seite!

- b) Wählen Sie zwei falsche Aussagen aus der vorherigen Tabelle aus und berichtigen Sie diese in jeweils einem Satz.

Teilaufgabe II: Datenbanken

Aufgabe 1 (Vermischte Fragen)

(21 Punkte)

Beantworten Sie die folgenden Fragen und begründen oder erläutern Sie Ihre Antwort.

- a) Erläutern Sie die Begriffe Kardinalität und Partizipität. Welche Arten von Partizipität gibt es in der ER-Modellierung? Nennen und erklären Sie diese kurz.
- b) Mit welchen beiden Befehlen kann eine Transaktion beendet werden? Nennen Sie diese und erklären Sie den Unterschied.
- c) Erläutern Sie den Unterschied zwischen einer kurzen und einer langen Sperre.
- d) Stellen Sie außerdem die Kompatibilitätsmatrix zur Umsetzung des ACID-Prinzips mit den richtigen Werten dar. S stehe dabei für eine Lese- und X für eine Schreibsperre.
- e) Nennen und erklären Sie kurz die Armstrong-Axiome. Sind diese vollständig und korrekt?
- f) Was versteht man unter einem (Daten-) Katalog (Data Dictionary) und was enthält dieser (es genügt eine Auswahl zu nennen)?
- g) Erklären Sie das konservative und das strikte Zwei-Phasen-Sperrprotokoll.
- h) Erklären Sie die Begriffe „Steal/NoSteal“ und „Force/NoForce“ im Kontext der Systempufferverwaltung eines DBS.

Aufgabe 2 (ER-Modellierung)

(35 Punkte)

Erstellen Sie ein möglichst einfaches ER-Schema, das alle gegebenen Informationen enthält. Attribute von Entitäten und Beziehungen sind anzugeben, Schlüsselattribute durch Unterstreichen zu kennzeichnen. Verwenden Sie für die Angabe der Kardinalitäten von Beziehungen die Min-Max-Notation. Führen Sie Surrogatschlüssel (künstlich definierte Schlüssel) nur dann ein, wenn es nötig ist, und modellieren Sie nur die im Text vorkommenden Elemente.

Zunächst gibt es Autos, die einen eindeutigen Namen, einen Typ sowie eine Liste an Ausstattungen besitzen. Autos werden aus Bauteilen zusammengesetzt. Diese besitzen eine ID sowie eine Beschreibung. Jedes Bauteil wird von genau einem Zulieferer geliefert. Zu jedem Zulieferer werden sein Name sowie seine E-Mail-Adresse gespeichert. Weiter gibt es Werke, die einen eindeutigen Namen sowie einen Standort besitzen. Jedes Werk besteht aus Hallen, welche werksintern eindeutig nummeriert sind. Zudem besitzt eine Halle noch eine gewisse Größe (in m²). Es gibt genau zwei Typen von Hallen: Produktionshallen und Ersatzteillager. In jeder Produktionshalle wird mindestens ein Auto hergestellt. Zu den Ersatzteillagern wird festgehalten, welche Bauteile und wie viele davon sich dort befinden.

Fortsetzung nächste Seite!

Zu jedem Mitarbeiter werden eine eindeutige ID, sein Vor- und Nachname, die Adresse (Straße, PLZ, Ort), das Gehalt sowie das Geschlecht gespeichert. Mitarbeiter werden unter anderem in Reinigungskräfte, Werksarbeiter und Ingenieure unterteilt. Zu den Ingenieuren wird zusätzlich der Hochschulabschluss gespeichert. Ingenieure sind genau einem Werk zugeordnet, Werksarbeiter einer Halle. Eine Reinigungskraft reinigt mindestens eine Halle. Jede Halle muss regelmäßig gereinigt werden. Weiter sind Ingenieure Projekten zugeteilt. Zudem wird zu jedem Projekt genau ein Ingenieur als Projektleiter festgehalten.

Aufgabe 3 (Relationale Algebra)**(8 Punkte)**

a) Gegeben seien die folgenden beiden Relationen R1 und R2:

R1	P	Q	S
	10	einfach	5
	15	b	8
	13	einfach	6

R2	A	B	C
	10	b	6
	13	c	3
	10	b	5

Geben Sie das Ergebnis des folgenden relationalen Ausdrucks an:

$$R_1 \bowtie_{R_1.P=R_2.A} R_2 \quad (\text{Equi-Join})$$

a) Zeichnen Sie den Operatorbaum zu folgender Abfrage in relationaler Algebra:

$$\begin{aligned} &\pi_{NAME, BERUF, PNAME} \left(\left(\pi_{ANR, NAME, BERUF} (PERSON) \right. \right. \\ &\quad \bowtie \pi_{ANR} (\sigma_{EIN > 1000000} (ABT))) \\ &\quad \left. \left. \bowtie \pi_{ANR, PNAME} (\sigma_{ORT = DD} (PROJ)) \right) \right) \end{aligned}$$

b) Ist der linke (bzw. rechte) Verbundoperator (Left- bzw. Right-Outer Join) assoziativ? Falls ja, beweisen Sie die Aussage, falls nein, geben Sie ein Gegenbeispiel an.

Aufgabe 4 (Normalisierung)**(22 Punkte)**

Gegeben ist das folgende Relationenschema in erster Normalform, bestehend aus zwei Relationen:

Relation1(A, B, C, D, E)

Relation2(F, G, H, A, E)

In diesem Schema gelten die folgenden funktionalen Abhängigkeiten:

- A B → C
- A B C → E
- A → D
- F G → H A
- G H → E

Fortsetzung nächste Seite!

- a) Nennen Sie die Bedingungen, damit ein Schema in erster Normalform ist.
- b) Überprüfen Sie, ob das Schema in zweiter Normalform ist.
- c) Wenden Sie den Synthesealgorithmus an, um das Schema in ein Schema in dritter Normalform zu überführen.
- d) Sei nun das Relationenschema $R(A,B,C,D)$ in erster Normalform gegeben. In R gelten die folgenden funktionalen Abhängigkeiten:
- $A \ B \rightarrow D$
 - $B \rightarrow C$
 - $C \rightarrow B$

Welches ist die höchste Normalform, in der sich das Schema R befindet? Begründen Sie Ihre Entscheidung.

Aufgabe 5 (Transaktionen)

(10 Punkte)

Gegeben sind die folgenden transaktionsähnlichen Abläufe. (Zunächst wird auf das Setzen von Sperren verzichtet.) Hierbei steht $R(X)$ für ein Lesezugriff auf X und $W(X)$ für einen Schreibzugriff auf X .

<p>T1:</p> <p>$R(A)$ $A := A-10$ $W(A)$ $R(C)$ $R(B)$ $B = B+10$ $W(B)$</p>	<p>T2:</p> <p>$R(D)$ $D := D-20$ $W(D)$ $R(A)$ $A = A+20$ $W(A)$</p>
--	---

Betrachten Sie folgenden Schedule:

T1	T2
$R_1(A)$	
	$R_2(D)$
	$D := D-20$
	$W_2(D)$
	$R_2(A)$
	$A := A+20$
	$W_2(A)$
$A := A-10$	
$W_1(A)$	
$R_1(C)$	
$R_1(B)$	
$B = B+10$	
$W_1(B)$	

Fortsetzung nächste Seite!

- a) Geben Sie die Werte von A, B, C und D nach Ablauf des Schedules an, wenn mit A = 100, B = 200, C = true und D = 150 begonnen wird.
- b) Geben Sie den Dependency-Graphen des Schedules an.
- c) Geben Sie alle auftretenden Konflikte an.
- d) Begründen Sie, ob der Schedule serialisierbar ist.
- e) Beschreiben Sie, wie die beiden Transaktionen mit LOCK Aktionen erweitert werden können, so dass nur noch serialisierbare Schedules ausgeführt werden können. Die Angabe eines konkreten Schedules ist nicht zwingend notwendig.

Aufgabe 6 (SQL)**(18 Punkte)**

Gegeben ist folgendes Relationenschema zur Verwaltung von Daten aus der Fußballweltmeisterschaft:

```
NATION (Land, Kapitän, Trainer)
    Kapitän ist Fremdschlüssel zu Spieler_ID in SPIELER.
MATCH (Match_ID, Ort, Datum, Team1, Team2, ToreTeam1, ToreTeam2)
    Team1 ist Fremdschlüssel zu Land in NATION.
    Team2 ist Fremdschlüssel zu Land in NATION.
SPIELER (Spieler_ID, Name, Vorname, Wohnort, Land)
    Land ist Fremdschlüssel zu Land in NATION.
PLATZVERWEISE (Platzverweis_ID, Match_ID, Spieler_ID, Spielminute)
    Match_ID ist Fremdschlüssel zu Match_ID in MATCH.
    Spieler_ID ist Fremdschlüssel zu Spieler_ID in SPIELER.
```

Die Primärschlüssel der Relationen sind wie üblich durch Unterstreichen gekennzeichnet. Pro Ort und Datum findet jeweils nur ein Spiel statt.

Formulieren Sie folgende Abfragen in SQL. Vermeiden Sie nach Möglichkeit übermäßige Nutzung von Joins und Views.

- a) Ermitteln Sie die Anzahl der Platzverweise pro Spieler und geben Sie jeweils Name und Vorname des Spielers mit aus. Die Ausgabe soll nach der Anzahl der Platzverweise absteigend erfolgen.
- b) Welches ist die maximale Anzahl an Toren, die eine Mannschaft insgesamt im Turnier erzielt hat? (Sie dürfen der Einfachheit halber annehmen, dass jede Mannschaft jeweils mindestens einmal als Team1 und Team2 angetreten ist.)
- c) Wie viele Tore sind im Turnier insgesamt gefallen?
- d) Ermitteln Sie die Namen und Länder der fünf Spieler, die nach der kürzesten Spielzeit einen Platzverweis erhielten. Die Ausgabe soll nummeriert erfolgen (beginnend bei 1 für die kürzeste Spielzeit).

Fortsetzung nächste Seite!

Aufgabe 7**(6 Punkte)**

Skizzieren Sie zunächst einen Anfragegraphen zu folgender SQL-Abfrage. Entscheiden Sie sich anschließend für eine mögliche Optimierung und zeichnen Sie den Anfragegraphen nach dieser Optimierung.

```
SELECT DISTINCT s.Semester
FROM Studenten s, hören h, Vorlesungen v, Professoren p
WHERE p.Name= 'Sokrates'
AND v.gelesenVon = p.PersNr
AND v.VorlNr = h.VorlNr
AND h.MatrNR = s.MatrNr
```

Thema Nr. 2**Teilaufgabe I: Softwaretechnologie****Aufgabe 1 (Projektmanagement)****(15 Punkte)**

Gegeben seien folgende Tätigkeiten mit ihren Abhängigkeiten und Dauern:

Task	Dauer (in h)	Abhängigkeiten
T1	3	/
T2	6	/
T3	2	T1
T4	2	T2
T5	5	T1
T6	3	T4, T5
T7	6	T3
T8	7	T4
T9	4	T6, T8
T10	1	T7, T9

- Zeichnen Sie ein CPM-Diagramm basierend auf der gegebenen Aufgabenliste. Benutzen Sie explizite Start- und Endknoten.
- Als *Slack* bezeichnet man die Zeit, um die eine Aufgabe bezüglich ihres frühesten Startzeitpunktes verzögert werden kann, ohne dass es Probleme bei der fristgerechten Fertigstellung des Projektes gibt. Berechnen Sie den Slack für alle Aktivitäten und ergänzen Sie ihn in Ihrem Diagramm.
- Zeichnen Sie den kritischen Pfad in Ihr Diagramm ein oder geben Sie die Tasks des kritischen Pfades in der folgenden Form an: **Start ! . . . ! Ende**. Sollte es mehrere kritische Pfade geben, geben Sie auch diese an. Wie lange ist die Dauer des kritischen Pfades bzw. der kritischen Pfade?

Aufgabe 2 (Entwicklungsprozesse)**(20 Punkte)**

- Erklären Sie den Unterschied zwischen iterativen und inkrementellen Entwicklungsprozessen. Nennen Sie zudem je ein Prozessmodell als Beispiel.
- Nennen und erklären Sie kurz die vier Leitsätze der agilen Softwareentwicklung laut Agilem Manifest.
- Beschreiben Sie die wesentlichen Aktivitäten in Scrum (agiles Entwicklungsmodell) inklusive deren zeitlichen Ablaufs. Gehen Sie dabei auch auf die Artefakte ein, die im Verlauf der Entwicklung erstellt werden.
- Nennen und erklären Sie die Rollen in einem Scrum-Team.

Fortsetzung nächste Seite!

Aufgabe 3 (Anforderungsanalyse)**(20 Punkte)**

Erstellen Sie ein Use-Case Diagramm zu einem Webshop mit den Akteuren Kunde, Premium-Kunde und Angestellte des Webshops. Dabei sind folgende Anforderungen zu beachten:

- /F010/ Ein Kunde bzw. Premium-Kunde kann nach Artikeln suchen.
- /F020/ Ein Kunde bzw. Premium-Kunde kann Artikel ansehen.
- /F030/ Ein Kunde bzw. Premium-Kunde kann für sich Artikel bestellen.
- /F040/ Die Bestellung eines Artikels beinhaltet eine Bestellprüfung durch einen Angestellten des Webshops.
- /F050/ Die Bestellung eines Artikels inkludiert die Eingabe einer Lieferadresse.
- /F060/ Die Bestellung eines Artikels umfasst auch die Bezahlung. Dies inkludiert die Auswahl einer Bezahlart.
- /F070/ Als Bezahlart werden Kreditkarte, Überweisung und Kauf auf Rechnung akzeptiert.
- /F080/ Die Bezahlung (Bezahlart) mittels Paypal wird nicht unterstützt.
- /F090/ Ein Premium-Kunde kann Rabattcodes einlösen.
- /F100/ Ein Premium-Kunde kann Artikel reservieren.
- /F110/ Das Reservieren eines Artikels inkludiert ebenso die Auswahl einer Bezahlart.

Hinweis: Verwenden Sie, falls sinnvoll, Vererbung zwischen den Rollen sowie den Anwendungsfällen.

Aufgabe 4 (Entwurfsmuster)**(20 Punkte)**

- a) Nennen und erklären Sie kurz die drei Kategorien der Organisation von klassischen Entwurfsmustern. Geben Sie zu jeder Kategorie ein Beispiel-Pattern an.
- b) Erstellen Sie ein Klassendiagramm zu den Komponenten einer Beobachtungsstation, welche aus einem einzigen Wasserstandsmesser besteht. Dabei sollen bei Änderungen des Wasserstandes der aktuelle Wasserstand sowohl auf der Konsole als auch in eine Log-Datei geschrieben werden. Der momentane Wasserstand soll dabei mittels einer Variablen des Datentyps `double` dargestellt werden. Die verschiedenen Anzeigearten (Konsolenanzeige, Logger) sollen als verschiedene Klassen modelliert werden und enthalten jeweils nur eine Methode zum Anzeigen bzw. Schreiben des aktuellen Wasserstandes. Verwenden Sie für die Realisierung dieses Klassendiagramms die passenden Entwurfsmuster.

Hinweise:

- Getter und Setter müssen nicht eingezeichnet werden.
- Fügen Sie auch Methoden ein, welche durch die einzelnen Klassen implementiert werden.

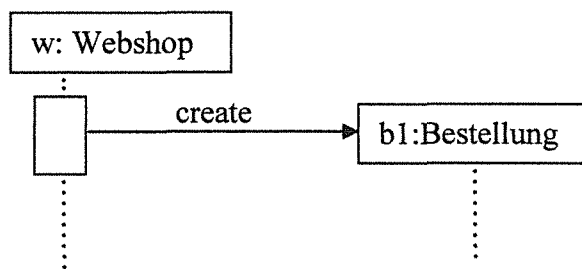
Fortsetzung nächste Seite!

Aufgabe 5 (Implementierung)**(20 Punkte)**

- Nennen Sie vier Programmierparadigmen.
- Erläutern Sie die Begriffe *Overloading* und *Overriding*, sowie deren Unterschiede.
- Erläutern Sie, wie sich zentrale und dezentrale Versionsverwaltung unterscheiden.
- Erstellen Sie ein Sequenzdiagramm zur Methode `main` der Klasse `Webshop`.

Hinweise:

- Arithmetische Operationen müssen nicht weiter aufgelöst werden.
- Listenoperationen müssen nicht explizit dargestellt werden.
- Auf das Zeichnen einer passiven Lebenslinie muss nicht geachtet werden.
- Übertragen Sie das untenstehende Diagramm als Ausgangspunkt in Ihren Bearbeitungsbogen.



```
1 public class Webshop {
2     public static void main(String[] args) {
3
4         Bestellung b1 = new Bestellung();
5
6         // ab hier soll modelliert werden
7         Artikel a1 = new Artikel();
8         a1.setName("Taschenrechner");
9         a1.setPrice(10);
10
11        b1.addArticle(a1);
12
13        Bestellung b2 = new Bestellung();
14
15        Artikel a2 = new Artikel();
16        a2.setName("Lineal");
17        a2.setPrice(2.5);
18
19        Artikel a3 = new Artikel();
20        a3.setName("Bleistift");
21        a3.setPrice(0.7);
22
23        b2.addArticle(a3);
24
25        b1.addArticle(a2);
26
27        b1.getSize();
28
29        b2.getPrice();
30    }
31 }
32
33
```

Fortsetzung nächste Seite!

```
1 public class Artikel {
2
3     private String name;
4     private double price;
5
6     public void setName(String name) {
7         this.name = name;
8     }
9
10    public void setPrice(double price) {
11        this.price = price;
12    }
13
14    public double getPrice() {
15        return price;
16    }
17 }
```

```
1 import java.util.List;
2
3 public class Bestellung {
4
5     private List<Artikel> articles;
6     //Anzahl an Artikeln
7     private int size = 0;
8     //Gesamtpreis der Bestellung
9     private double price = 0;
10
11    public Bestellung() {
12        articles = new ArrayList<>();
13    }
14
15    public void addArticle(Artikel article) {
16        //muss nicht weiter aufgelöst werden, siehe Hinweise
17        articles.add(article);
18        size++;
19        //muss nicht weiter aufgelöst werden, siehe Hinweise
20        price = article.getPrice() + price;
21    }
22
23    public int getSize() {
24        return size;
25    }
26
27    public double getPrice() {
28        return price;
29    }
30 }
```

Aufgabe 6 (Refactoring)**(10 Punkte)**

- a) Erklären Sie die Begriffe Code-Smell und Refactoring, und erläutern Sie, wie diese zusammenhängen.
- b) Wenden Sie, sofern sinnvoll, ausgehend vom untenstehenden Diagramm verschiedene Refactoringschritte an (mindestens drei). Benennen Sie die angewandten Refactoring-Schritte und beschreiben Sie diese kurz in je einem Satz. Zeichnen Sie das Klassendiagramm für das Endresultat in Ihre Bearbeitung. Änderungen an öffentlichen Methoden und Attributen dürfen vorgenommen werden. Wenn Sie die Sichtbarkeit eines Attributs verändern, fügen Sie gegebenenfalls Getter und Setter für dieses Attribut ein.

PKW
- colour: String
- kilometres: int
+do_something_fancy(): void

LKW
- lk_w_coulor: String
- kilometres: int

```
1 public class PKW {  
2     private String colour;  
3     private int kilometres;  
4  
5     public void do_something_fancy {  
6         //Inhalt dieser Methode für die Aufgabe irrelevant  
7     }  
8 }
```

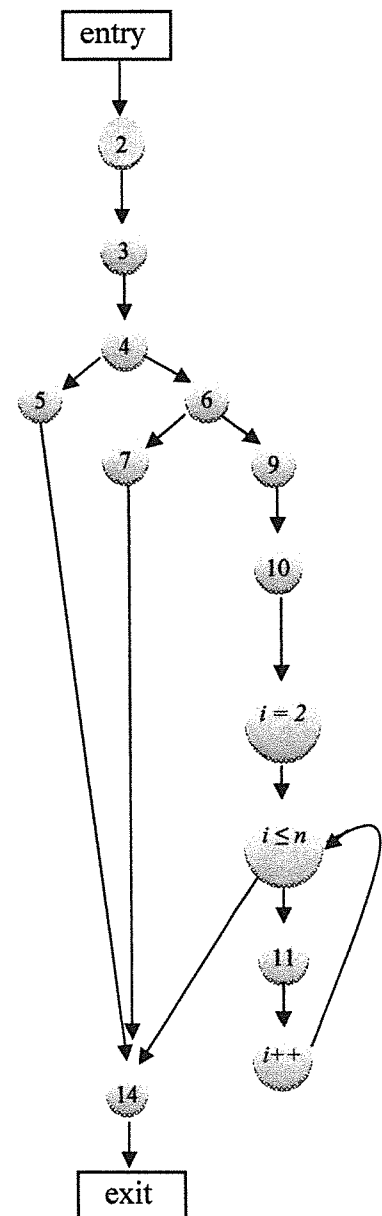
```
1 public class LKW {  
2     private String lk_w_coulor;  
3     private int kilometres;  
4 }
```

Fortsetzung nächste Seite!

Aufgabe 7 (Testing)**(15 Punkte)**

- Kann man durch Testen sicherstellen, dass ein Programm korrekt läuft? Begründen Sie Ihre Antwort.
- Nennen Sie fünf Qualitätsmerkmale von Softwareprodukten.
- Erklären Sie die Begriffe Verifikation und Validierung.
- Betrachten Sie die nachfolgende Java-Methode `foo` und deren Kontrollflussgraphen. Die Zahlen in den Knoten des Graphen verweisen auf die Zeilennummern im Programm.

```
1 public int foo(int n) {  
2     int N = n;  
3     int x;  
4     if (N < 0) {  
5         x = -1;  
6     } else if (N <= 1) {  
7         x = 1;  
8     } else {  
9         x = 1;  
10        for (int i = 2; i <= N; i++) {  
11            x = x * i;  
12        }  
13    }  
14    return x;  
15 }
```

**Fortsetzung nächste Seite!**

- i) Wie groß ist die zyklomatische Komplexität (cyclomatic complexity) der Methode `foo`? Geben Sie hierzu die Formel sowie auch Ihre Berechnungen mit an.
- ii) Bestimmen Sie die Anweisungsüberdeckung für die Funktionsaufrufe `foo(-1)` und `foo(2)`. Geben Sie jeweils die allgemeine Formel zur Berechnung der Überdeckung an, entnehmen Sie die konkreten Werte dieser Aufrufe aus dem Kontrollflussgraphen und berechnen Sie schließlich die erreichte Überdeckung. Brüche brauchen dabei nicht gekürzt werden. Beziehen Sie *entry* und *exit* Knoten in Ihre Berechnung ein.
- iii) Bestimmen Sie die Zweigüberdeckung für die Funktionsaufrufe `foo(-2)` und `foo(2)`. Geben Sie jeweils die allgemeine Formel zur Berechnung der Überdeckung an, entnehmen Sie die konkreten Werte dieser Aufrufe aus dem Kontrollflussgraphen und berechnen Sie schließlich die erreichte Überdeckung. Brüche brauchen dabei nicht gekürzt werden. Beziehen Sie *entry* und *exit* Knoten in Ihre Berechnung ein.

Teilaufgabe II: Datenbanken

Aufgabe 1 (Vermischte Fragen)

(12 Punkte)

Beantworten Sie die folgenden Fragen und begründen oder erläutern Sie Ihre Antwort in jeweils ein bis zwei Sätzen.

- a) Kann ein Tupel mehrfach im Ergebnis einer SQL-Anfrage enthalten sein?
- b) Was ist der Unterschied zwischen einem INNER JOIN und einem OUTER JOIN?
- c) Welche Auswirkung hat die Verwendung von ON DELETE CASCADE bei einem Fremdschlüsselattribut?
- d) Kann eine abgebrochene (*aborted*) Transaktion wieder fortgesetzt werden?
- e) Was versteht man unter einer *stored procedure* im Kontext einer Programmierschnittstelle für relationale Datenbanken (z.B. JDBC)?
- f) Was sind *check constraints* und wie wirken sich diese aus?

Aufgabe 2 (ER-Modellierung)

(26 Punkte)

Im Folgenden finden Sie die Beschreibung eines Online-Marktplatzes. Erstellen Sie zu dieser Beschreibung ein erweitertes ER-Diagramm. Kennzeichnen Sie die Primärschlüssel durch passendes Unterstreichen und geben Sie die Kardinalitäten in Chen-Notation (= Funktionalitäten) an. Kennzeichnen Sie auch die totale Teilnahme von Entity-Typen an Beziehungstypen.

Es gibt Produkte. Diese haben eine eindeutige Bezeichnung, einen Beschreibungstext und eine Bewertung. Außerdem gibt es Personen, die entweder Kunde, Händler oder beides sind. Jede Person hat einen Nachnamen, einen oder mehrere Vornamen, ein Geburtsdatum und eine E-Mail-Adresse, mit der diese eindeutig identifiziert werden kann.

Fortsetzung nächste Seite!

Das System verwaltet außerdem Zahlungsmittel. Jedes Zahlungsmittel ist entweder eine Kreditkarte oder eine Bankverbindung für Lastschriften. Für das Lastschriftverfahren wird die international eindeutige IBAN und der Name des Kontoinhabers erfasst, bei Zahlung mit Kreditkarte der Name des Karteninhabers, die eindeutige Kartennummer, das Ablaufdatum sowie der Kartenanbieter. Es gibt Transaktionen. Jede Transaktion bezieht sich stets auf ein Produkt, einen Kunden, einen Händler und auf ein Zahlungsmittel, das für die Transaktion verwendet wird. Jede Transaktion enthält außerdem den Preis, auf den sich Kunde und Händler geeinigt haben, das Abschlussdatum sowie eine Lieferadresse, an die das Produkt versandt wird.

Aufgabe 3 (Relationaler Entwurf)

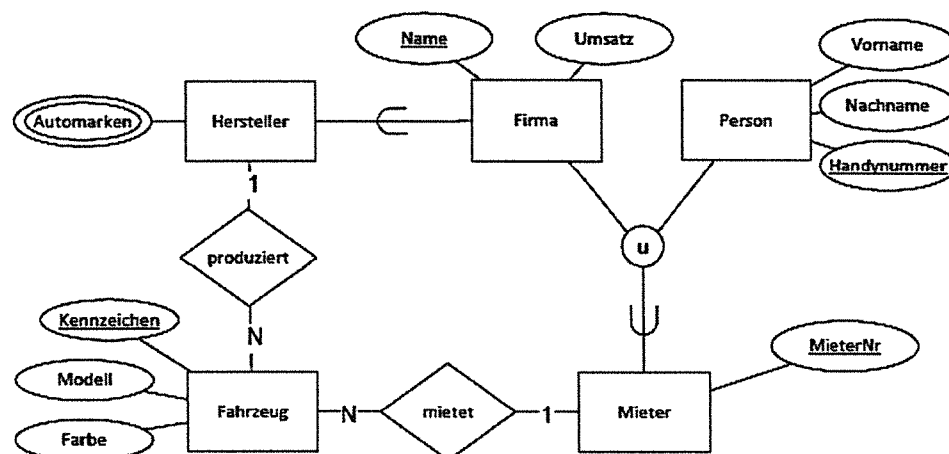
(20 Punkte)

Entwerfen Sie zum untenstehenden ER-Diagramm ein Relationenschema (in dritter Normalform, 3. NF) mit möglichst wenigen Relationen.

Verwenden Sie dabei folgende Notation: Primärschlüssel werden durch Unterstreichen gekennzeichnet, Fremdschlüssel durch die Nennung der Relation, auf die sie verweisen, in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert. Wenn ein Attribut zur korrekten Abbildung des ER-Diagramms als **UNIQUE** oder **NOT NULL** ausgezeichnet werden muss, geben Sie dies an.

Beispiel:

```
Relation1 (Primärschlüssel, Attribut1, Attribut2,
Fremdschlüsselattribut1[Relation1],
(Fremdschlüssel2_Attribut1, Fremdschlüssel2_Attribut2)[Relation2]);
Attribut1 UNIQUE
Attribut2 NOT NULL
```



Fortsetzung nächste Seite!

Aufgabe 4 (SQL)**(34 Punkte)**

Gegeben sind folgende Relationen:

Mitarbeiter(MitarbeiterID, Vorname, Nachname, Adresse, Gehalt,
Vorgesetzter[Mitarbeiter], AbteilungsID[Abteilung])
Vorgesetzter NOT NULL
Abteilung(AbteilungsID, Bezeichnung)
Bezeichnung UNIQUE NOT NULL

Verwenden Sie im Folgenden nur Standard-SQL und keine produktspezifischen Erweiterungen. Sie dürfen bei Bedarf Views anlegen. Geben Sie einen Datensatz nicht mehrfach aus.

- a) Schreiben Sie eine SQL-Anweisung, die die Tabelle *Mitarbeiter* anlegt. Gehen Sie davon aus, dass die Tabelle *Abteilung* bereits existiert.
- b) Schreiben Sie eine SQL-Anweisung, die Vor- und Nachnamen der Mitarbeiter der Abteilung mit der Bezeichnung *Vertrieb* ausgibt, absteigend sortiert nach MitarbeiterID.
- c) Schreiben Sie eine SQL-Anweisung, die Vor- und Nachnamen sowie das Gehalt von Mitarbeitern ausgibt, die mehr verdienen als ihr Chef. Sortieren Sie die Ausgabe absteigend nach dem Gehalt.
- d) Schreiben Sie eine SQL-Anweisung, die das Gehalt von allen Mitarbeitern aus der Abteilung mit der ID 42 um 10% erhöht.
- e) Schreiben Sie eine SQL-Anweisung, welche den Vornamen, die Nachnamen und das Gehalt der sieben bestbezahlten Mitarbeiter aus der Buchhaltung ausgibt. Standardkonforme Sprachkonstrukte, die eine Beschränkung der Ausgabe bewirken, sind erlaubt.
- f) Schreiben Sie eine SQL-Anweisung, die für jede Abteilung die Mitarbeiter ermittelt, die am wenigsten verdienen. Dabei sollen Vorname, Nachname und die Abteilungsbezeichnung der Mitarbeiter ausgegeben werden.

Aufgabe 5 (Relationale Algebra)**(10 Punkte)**

Formulieren Sie basierend auf den in der letzten Aufgabe gegebenen Relationen die geforderten Anfragen **in der Relationenalgebra**.

- a) Formulieren Sie eine Anfrage, welche die Vornamen und Nachnamen der Mitarbeiter ausgibt, die in der Buchhaltung arbeiten.
- b) Formulieren Sie eine Anfrage, welche die Vornamen und Nachnamen der Mitarbeiter ausgibt, die in keiner Abteilung arbeiten.

Fortsetzung nächste Seite!

Aufgabe 6 (Normalformen)**(8 Punkte)**

Gegeben ist die Relation Prüfung (Prüfungsnummer, Fakultät, Prüfungsname, Dozent, Prüfungstyp, ECTS) mit den beiden Schlüsselkandidaten (Prüfungsnummer, Fakultät) und (Fakultät, Prüfungsname, Dozent).

Alle Attributwerte sind atomar. Es gelten nur die durch die Schlüsselkandidaten implizierten funktionalen Abhängigkeit.

Geben Sie die höchste Normalform an, die die Relation Prüfung erfüllt. Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind und dass mindestens eine Bedingung der nächsthöheren Normalform verletzt ist. Beziehen Sie sich bei der Begründung auf die gegebene Relation und nennen Sie nicht nur die allgemeinen Definitionen der Normalformen.

Aufgabe 7 (Optimierung)**(10 Punkte)**

- a) Erläutern Sie kurz, was Indizes sind und warum diese in Datenbanksystemen verwendet werden.
- b) Übertragen Sie folgendes SQL-Statement in einen *nicht optimierten* algebraischen Term oder in einen Anfragegraphen.

```
SELECT Kunde.Name , Kunde.Geburtsdatum  
FROM Kunde , Rechnung  
WHERE Kunde.ID = Rechnung.Kunde  
AND Rechnung.Summe < 100;
```

- c) Nennen Sie zwei Möglichkeiten, den algebraischen Term bzw. den Anfragegraphen aus der vorhergehenden Teilaufgabe logisch (d. h. algebraisch) zu optimieren. Beziehen Sie sich auf konkrete Stellen und Operatoren des von Ihnen aufgestellten algebraischen Ausdrucks.