Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Herbst 2020

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: 25

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1 (Verifikation) [30 PUNKTE]

a) Definieren Sie die Begriffe „partielle Korrektheit" und „totale Korrektheit" und grenzen Sie sie voneinander ab.

b) Geben Sie die Verifikationsregel für die abweisende Schleife **while**(B) A an.

c) Erläutern Sie kurz und prägnant die Schritte zur Verifikation einer abweisenden Schleife mit Vorbedingung P und Nachbedingung Q.

d) Wie kann man die Terminierung einer Schleife beweisen?

e) Geben Sie für das folgende Suchprogramm die nummerierten Zusicherungen an. Lassen Sie dabei jeweils die invariante Vorbedingung P des Suchprogramms weg. Schreiben Sie nicht auf dem Aufgabenblatt!

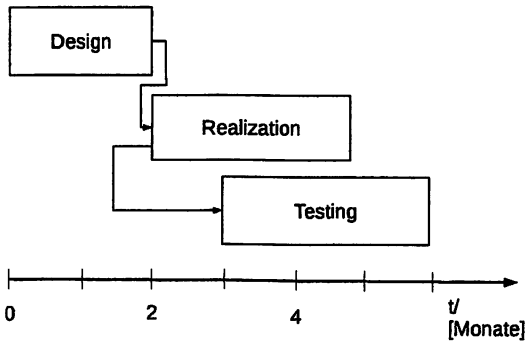
```
// P ≡ n > 0 ∧ a_0 ... a_{n-1} ∈ Z^n ∧ m ∈ Z
i = -1;
// (1)
j = 0;
// (2)
while (i == -1 && j < n) // (3)
{     // (4)
      if (a[j] == m){
            // (5)
            i = j;
            // (6)
      }
      else {
            // (7)
            j = j + 1;
            // (8)
      }
      // (9)
}
// Q ≡ P ∧ (i = -1 ∧ ∀ 0 ≤ k < n : a_k ≠ m) ∨ (i ≥ 0 ∧ a_i = m)
```

Fortsetzung nächste Seite!

Aufgabe 2 (Projektplanung) [15 PUNKTE]

Die Planung eines Softwareprojekts kann z. B. in Form von Gantt-Diagrammen oder CPM-Netzwerken (kritischer Pfad Methode) festgehalten werden.

Folgendes Gantt-Diagramm zeigt einen Teil der Projektplanung in einem klassischen Softwareentwicklungsprozess:

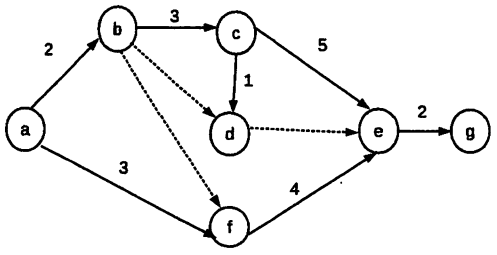


(a) Im Diagramm werden 3 Phasen aus dem klassischen Softwareentwicklungsprozess genannt. Welche Phase sollte dem Design (Entwurf) immer vorangehen?

(b) Wandeln Sie das Gantt-Diagramm in ein CPM-Netzwerk um. Fügen Sie dazu einen zusätzlichen Start- und Endknoten hinzu. Das Ende des Projekts ist durch das Ende aller Aktivitäten bedingt.

(c) Welche im obigen Gantt-Diagramm nicht enthaltenen Beziehungsarten zwischen Aktivitäten können in einem Gantt-Diagramm noch auftreten? Nennen Sie auch deren Bedeutung.

Gegeben sei nun das folgende CPM-Netzwerk:



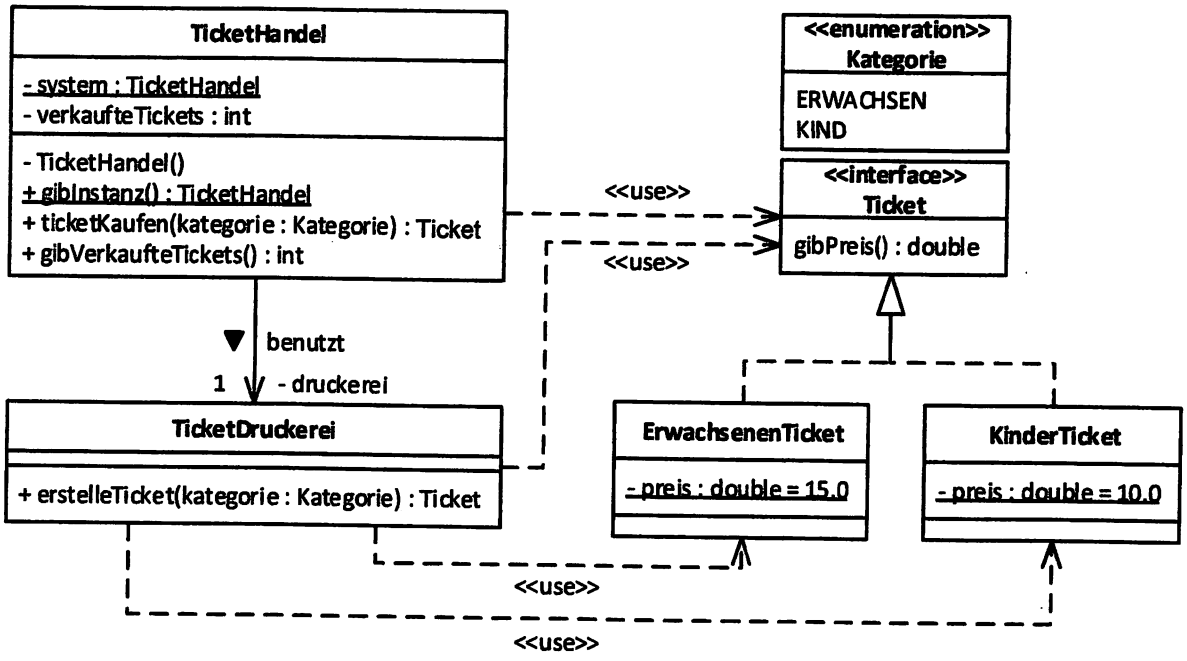
(d) Geben Sie für jedes Ereignis die früheste Zeit an.

(e) Geben Sie für jedes Ereignis die späteste Zeit an.

Fortsetzung nächste Seite!

(f) Geben Sie einen kritischen Pfad durch das Netz an! Wie wirkt sich eine Verzögerung von 5 Zeiteinheiten auf dem kritischen Pfad auf das Projektende aus?

Aufgabe 3 (Entwurfsmuster) [30 PUNKTE]



Ihnen sei ein **UML Klassendiagramm** zu folgendem Szenario gegeben. Ein Benutzer (nicht im Diagramm enthalten) kann über einen `TicketHandel` Tickets erwerben. Dabei muss der Benutzer eine der zwei Ticketkategorien angeben. Das Handelsystem benutzt eine `TicketDruckerei`, um ein passendes Ticket für den Benutzer zu erzeugen.

a) Im angegebenen Klassendiagramm wurden **zwei** unterschiedliche Entwurfsmuster verwendet. Um welche Muster handelt es sich? Geben Sie jeweils den Namen des Musters sowie die Elemente des Klassendiagramms an, mit denen diese Muster im Zusammenhang stehen. ACHTUNG: Es handelt sich dabei **nicht** um das *Interface*- oder das Vererbungsmuster.

b) Nennen Sie zwei generelle Vorteile von Entwurfsmustern.

c) Geben Sie eine Implementierung der Klasse `TicketHandel` an. Bei der Methode `ticketKaufen()` wird die Anzahl der verkauften Tickets um 1 erhöht und ein entsprechendes Ticket erstellt und zurückgegeben. Beachten Sie den Hinweis auf der nächsten Seite.

d) Geben Sie eine Implementierung der Klasse `TicketDruckerei` an.

e) Geben Sie eine Implementierung der Klasse `KinderTicket` an.

Fortsetzung nächste Seite!

Hinweis: Die Implementierungen **müssen** sowohl dem Klassendiagramm, als auch den Konzepten der verwendeten Muster entsprechen. Verwenden Sie eine objektorientierte Programmiersprache, vorzugsweie **Java**. Sie müssen sich an der nachfolgenden Testmethode und ihrer Ausgabe orientieren. Die Testmethode muss mit Ihrer Implementierung ausführbar sein und sich semantisch korrekt verhalten.

Quelltext der Testmethode:

```
1 public static void main(String[] args) {
2     TicketHandel.gibInstanz().ticketKaufen(Kategorie.ERWACHSEN);
3     TicketHandel.gibInstanz().ticketKaufen(Kategorie.KIND);
4     System.out.println (" Anzahl verkaufter Tickets: "
5         + TicketHandel.gibInstanz().gibVerkaufteTickets());
6 }
```

Konsolenausgabe:

```
Anzahl verkaufter Tickets: 2
```

Aufgabe 4 (White-Box-Testverfahren) [30 PUNKTE]

Diese Aufgabe behandelt *Wortpalindrome*, also Wörter, die vorwärts und rückwärts gelesen jeweils dieselbe Zeichenkette bilden, z. B. `Otto` oder `Rentner`. Leere Wortpalindrome (also Wortpalindrome der Wortlänge 0) sind dabei nicht zulässig.

Folgende **Java-Methode** prüft, ob das übergebene Zeichen-Array ein Wortpalindrom darstellt:

```
1  public static boolean istWortpalindrom(char[] wort) {
2      boolean resultat = false;
3      if (wort != null) {
4          int laenge = wort.length;
5          if (laenge >= 2) {
6              resultat = true;
7              for (int i = 0; i < laenge/2; ++i) {
8                  char c1 = wort[i];
9                  char c2 = wort[laenge-1-i];
10                 if (Character. toLowerCase (c1) != Character. toLowerCase (c2))
11                 {...}
12                     resultat = false;
13                     break;
14                 }
15             }
16         } else {
17             if (laenge == 1) {
18                 resultat = true;
19             }
20         }
21     }
22     return resultat;
23 }
```

Fortsetzung nächste Seite!

(a) Geben Sie für die Methode einen **Kontrollflussgraphen** an, wobei Sie die Knoten mit den jeweiligen Zeilennummern im Quelltext beschriften.

(b) Geben Sie eine **minimale Testmenge** an, die das Kriterium der **Anweisungsüberdeckung** erfüllt.
Hinweis: Eine *Testmenge* ist *minimal*, wenn es keine Testmenge mit einer kleineren Zahl von Testfällen gibt. Die Minimalität muss nicht bewiesen werden.

(c) Geben Sie eine **minimale Testmenge** an, die das Kriterium der **Boundary-Interior-Pfadüberdeckung** erfüllt.
Hinweis: Das Kriterium *Boundary-Interior-Pfadüberdeckung* beschreibt einen Spezialfall der Pfadüberdeckung, wobei nur Pfade berücksichtigt werden, bei denen jede Schleife nicht mehr als zweimal durchlaufen wird.

(d) Im Falle des Kriteriums Pfadüberdeckung können minimale Testmengen sehr groß werden, da die Anzahl der Pfade sehr schnell zunimmt. Wie viele **mögliche Pfade** ergeben sich maximal für eine Schleife, die drei einseitig bedingte Anweisungen hintereinander enthält und bis zu zweimal durchlaufen wird? Geben Sie Ihren Rechenweg an (das Ergebnis alleine gibt keine Punkte).

(e) Könnte für das hier abgebildete Quelltext-Beispiel auch das Verfahren der **unbegrenzten Pfadüberdeckung** (also Abdeckung aller möglicher Pfade ohne Beschränkung) als Test-Kriterium gewählt werden? Begründen Sie.

Aufgabe 5 (Lebenszyklus) [15 PUNKTE]

(a) Nennen Sie fünf kritische Faktoren, die bei der Auswahl eines Vorgehensmodells helfen können und ordnen Sie plangetriebene und agile Prozesse entsprechend ein.

(b) Nennen und beschreiben Sie kurz die Rollen im Scrum.

(c) Nennen und beschreiben Sie drei Scrum Artefakte. Nennen Sie die verantwortliche Rolle für jedes Artefakt.

(d) Beschreiben Sie kurz, was ein Sprint ist. Wie lange sollte ein Sprint maximal dauern?

Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanksysteme

Aufgabe 1 (Entwurfstheorie: Entity-Relationship-Diagramm) [19 PUNKTE]

Sie sollen für einen Zoo eine Datenbank modellieren, welche folgende Daten und Zusammenhänge beinhaltet:

Der Zoo soll eine Reihe von Tieren unterschiedlicher Tierarten beherbergen. Für jede Tierart soll mindestens ein Gebäude existieren. Da es aber auch erwünscht ist, dass mehrere Tierarten in einem Gebäude zusammenleben, können in einem Gebäude auch mehrere unterschiedliche Tierarten leben. Jedes Tier hat genau ein bestimmtes Gebäude als feste "Heimat".

Jeder Pfleger kümmert sich um mehrere Tiere. Für jede Tierart ist ein Pfleger speziell beauftragt. Der beauftragte Pfleger kennt sich besonders gut mit dieser Art aus.

Jede Tierart frisst bestimmte Futtermittel. Es gibt Futtermittel, die von mehreren verschiedenen Arten gefressen werden können. Die Futtermittel werden in unterschiedlichen Gebäuden gelagert, wobei es für jedes Futtermittel genau ein festgelegtes Gebäude zur Lagerung gibt.

Erstellen Sie zur Modellierung des Zoos ein Entity-Relationship-Diagramm in nachfolgenden Schritten:

1. Geben Sie zunächst alle sinnvollen Entitäten des Zoos in Listenform an.
   Hinweis: 5 Entitäten sind gesucht.

2. Zeichnen Sie ein Entity-Relationship-Diagramm, das die in 1. identifizierten Entitäten und alle im Text genannten Beziehungen modelliert.
   Hinweis: Gesucht sind 7 Beziehungen.

3. Tragen Sie für jede Beziehung gültige Funktionalitätsangaben in Ihr Modell ein.

Fortsetzung nächste Seite!

Aufgabe 2 (Überführung in ein relationales Modell) [24 PUNKTE]

Gegeben sei die ER-Modellierung von Flugverbindungen in Abbildung 1.

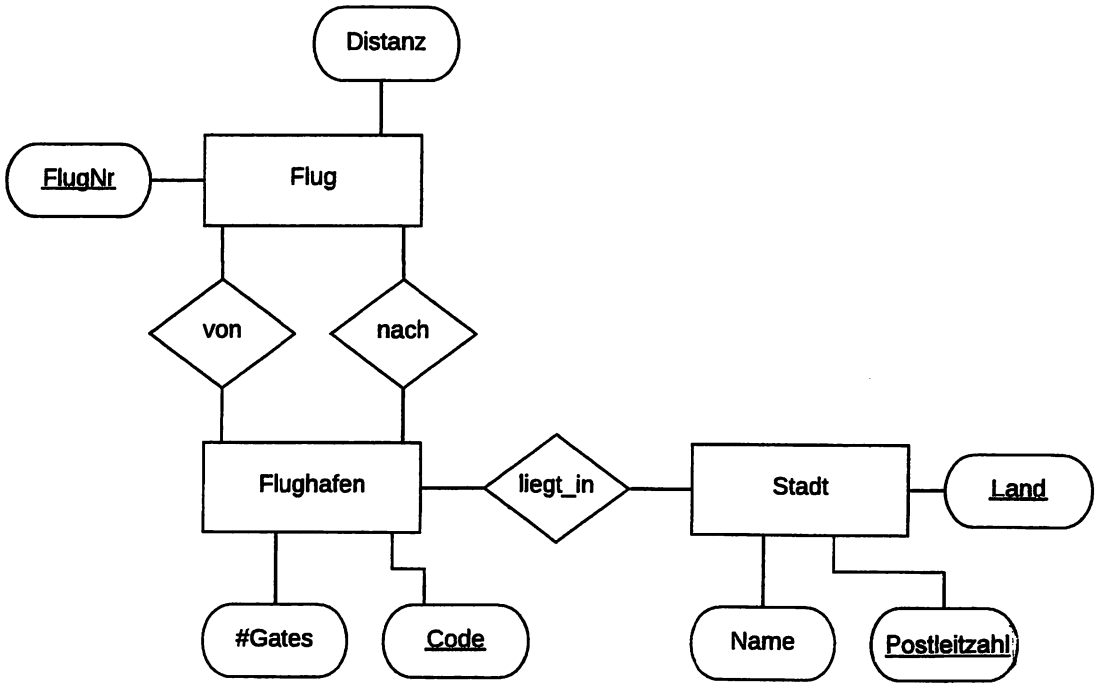


Abbildung 1: ER-Modellierung von Flügen

1. Geben Sie für alle Beziehungen (`von`, `nach`, `liegt_in`) sinnvolle Multiplizitäten an. Zeichnen Sie dazu das ER-Diagramm ab und tragen Sie die Multiplizitäten dort ein. Die Attribute (`FlugNr`, `Distanz`, ...) müssen hierbei nicht mit abgezeichnet werden. Beachten Sie, dass jeder Flug genau einen Start- und einen Zielflughafen haben soll.
2. Übertragen Sie das ER-Modell in ein relationales Schema unter Angabe sinnvoller Datentypen. Unterstreichen Sie die Schlüsselattribute jeder Relation. Eine Verfeinerung des relationalen Schemas ist noch nicht notwendig.
3. Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.

Fortsetzung nächste Seite!

Aufgabe 3 (Anfragesprachen) [15 PUNKTE]

Gegeben sei das Universitätsschema (eine beispielhafte Ausprägung befindet sich auf S. 13).

Hinweis: Beachten Sie, welche Anfragesprache jeweils gefordert ist!

1. Finden Sie Professoren, die **keine** Assistenten haben. Also die Professoren, für die es keinen Assistenten gibt, für die sie der "Boss" sind.

   Formulieren Sie die Anfrage im **Tupelkalkül**.

2. Finden Sie alle Studenten, die schon mindestens im 4. Semester sind und bereits die Vorlesung 'Wissenschaftstheorie' gehört haben.

   Formulieren Sie die Anfrage in **relationaler Algebra**.

3. Geben Sie einen Ausdruck an, der die Relation *noch_nicht_geprüft* erzeugt. Gesucht sind also alle (MatrNr, VorlNr)-Pärchen, zu denen es aktuell noch keinen entsprechenden Eintrag in der Relation *prüfen* gibt.

   Formulieren Sie die Anfrage in **relationaler Algebra**.

Aufgabe 4 (SQL) [23 PUNKTE]

Die folgenden Teilaufgaben beziehen sich auf das Universitätsschema, welches Sie in einer Beispielausprägung auf Seite 13 finden. Geben Sie für die folgenden Teilaufgaben jeweils eine SQL-Anfrage an.

Unter jeder Teilaufgabe finden Sie das erwartete Ergebnis für die Beispielausprägung. Ihre Anfragen müssen natürlich auch dann funktionieren, wenn die Ausprägung der Relationen anders ist als die Beispielausprägung.

1. Wie viele Studenten hören *'Logik'*?

count
1

2. In welchen Fächern ist die Durchschnittsnote schlechter als *1*? Geben Sie die Vorlesungsnummer (VorlNr) und den Titel aus.

VorlNr	Titel
4630	Die 3 Kritiken
5041	Ethik

Fortsetzung nächste Seite!

3. Geben Sie die Vorlesungsnummer (VorlNr) und den Titel aller Vorlesungen an, die mindestens eine Studentin bzw. ein Student des sechsten oder höheren Semesters hört.

VorlNr	Titel
4052	Logik
5001	Grundzüge
5022	Glaube und Wissen

4. Geben Sie die PersNr und den Namen aller Professoren aus, die mehr Assistenten haben als Augustinus.

PersNr	Name
2125	Sokrates
2127	Kopernikus

5. Geben Sie eine SQL-Anfrage an, welche für alle Professoren ausgibt, wie viele Vorlesungen sie halten und wie viele verschiedene Studenten sie jeweils in ihren Vorlesungen lehren. Beachten Sie, dass **alle** Professoren im Ergebnis enthalten sein müssen. Geben Sie folgende Attribute aus: PersNr, Name, AnzVorlesungen, AnzStudenten.

PersNr	Name	AnzVorlesungen	AnzStudenten
2125	Sokrates	3	3
2126	Russel	3	1
2127	Kopernikus	0	0
2133	Popper	1	1
2134	Augustinus	1	2
2136	Curie	0	0
2137	Kant	2	4

Fortsetzung nächste Seite!

Aufgabe 5 (Relationale Entwurfstheorie) [24 PUNKTE]

1. Geben Sie die höchste Normalform an, die für die angegebene Relation und FDs/MVDs noch gilt:

   a) $R : \{[A, B, C, D, E]\}$
   $BCD \rightarrow A$
   $D \rightarrow E$

   b) $R : \{[A, B, C, D, E]\}$
   $DE \rightarrow AC$
   $AC \rightarrow BDE$
   $BE \twoheadrightarrow ACD$

2. Gegeben sei die folgende Ausprägung der Relation $R : \{[A, B, C, D, E]\}$ mit den Attributen $A \in \{1, 2, 3\}$, $B \in \{S, W, V\}$, $C \in \{x, y, z\}$, $D \in \{I, M, P\}$ und $E \in \{\triangle, \blacksquare, \times\}$:

A	B	C	D	E
1	S	y	I	■
2	V	z	I	×
3	W	x	M	△
3	W	x	P	△
3	W	y	M	■
3	W	y	P	■

Ergänzen Sie die folgende MVD, sodass sie für die Ausprägung gilt und nicht-trivial ist:

$$AB \twoheadrightarrow$$

Fortsetzung nächste Seite!

3. Gegeben sei die Relation $R : \{[A, B, C, D, E]\}$ mit den FDs:

$$AB \rightarrow D$$
$$B \rightarrow AC$$
$$CD \rightarrow AB$$
$$DB \rightarrow C$$

Diese Relation hat die Kandidatenschlüssel BE und CDE. Überführen Sie sie mittels Synthesealgorithmus in die 3. Normalform. Geben Sie **alle Relationen** in der 3. Normalform an und **unterstreichen Sie in jeder einen Kandidatenschlüssel**. Geben Sie auch Ihr Zwischenergebnis jeweils nach der **Links-** und **Rechtsreduktion** an.

Aufgabe 6 (Anfragebearbeitung) [11 PUNKTE]

Die folgenden Teilaufgaben beziehen sich auf das Universitäts-Beispiel (s. Abb. 2, S. 13). Betrachten Sie die folgende SQL-Anfrage:

```
SELECT DISTINCT p.Name
FROM Professoren p, Vorlesungen v1, voraussetzen x, Vorlesungen v2
WHERE p.PersNr=v1.gelesenVon AND v1.VorlNr=x.Vorgänger AND
      x.Nachfolger=v2.VorlNr AND v2.Titel='Wissenschaftstheorie';
```

1. Was ermittelt diese Anfrage? Beschreiben Sie das Ergebnis in Worten.
2. Übersetzen Sie die SQL-Anfrage gemäß der kanonischen Übersetzung in die relationale Algebra (Operatorbaum-Darstellung). Etwaige Umbenennungsoperatoren brauchen dabei nicht angegeben werden.
3. Führen Sie nun die logische Optimierung durch. Gehen Sie hierbei von den Kardinalitäten und Selektivitäten einer typischen Universität (z. B. Ihrer Alma Mater) aus und dokumentieren Sie diese. Zeichnen Sie den Operatorbaum jeweils nach folgenden Schritten:
   a) Aufbrechen und Verschieben von Selektionen
   b) Zusammenfassung von Selektionen und Kreuzprodukten zu Joins
   c) Bestimmung/Optimierung der Joinreihenfolge

   "Pushing Projections", also das Einfügen und Verschieben von Projektionen, muss nicht durchgeführt werden.

Aufgabe 7 (Transaktionen) [4 PUNKTE]

Gegeben sei die folgende Historie H der Transaktionen T_1, T_2 und T_3:

$$r_1(a)\ w_1(b)\ r_2(b)\ r_2(a)\ r_2(c)\ w_2(c)\ r_3(c)\ w_1(d)\ c_1\ c_2\ c_3$$

1. Zeichnen Sie den Serialisierbarkeitsgraphen von H.
2. Ist die Historie serialisierbar? Wenn nein, warum nicht? Wenn ja, zu welcher seriellen Historie ist sie äquivalent?

Fortsetzung nächste Seite!

Beispielausprägung

Professoren

PersNr	Name	Rang	Raum
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Studenten

MatrNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Vorlesungen

VorlNr	Titel	SWS	gelesenVon
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

voraussetzen

Vorgänger	Nachfolger
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

hören

MatrNr	VorlNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022
29555	5001

Assistenten

PersNr	Name	Fachgebiet	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2134

prüfen

MatrNr	VorlNr	PersNr	Note
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2
25403	4630	2137	5

Abb. 2: Beispielausprägung für eine Universitäts-Datenbank

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1 (Entwicklungsprozesse) [26 PUNKTE]

a) Nennen Sie fünf Phasen, die im Wasserfallmodell durchlaufen werden sowie deren jeweiliges Ziel bzw. Ergebnis(-dokument).

b) Nennen Sie drei Arten der Softwarewartung und geben Sie jeweils eine kurze Beschreibung an.

c) Eine grundlegende Komponente des Extreme Programming ist „Continuous Integration". Erklären Sie diesen Begriff und warum man davon einen Vorteil erwartet.

d) Nennen Sie zwei Softwaremetriken und geben Sie jeweils einen Vor- und Nachteil an, der im Projektmanagement von Bedeutung ist.

e) Nennen und beschreiben Sie kurz drei wichtige Aktivitäten, welche innerhalb einer Sprint-Iteration von Scrum durchlaufen werden.

f) Erläutern Sie den Unterschied zwischen dem Product-Backlog und dem Sprint-Backlog.

g) Erläutern Sie, warum eine agile Entwicklungsmethode nur für kleinere Teams (max. 10 Personen) gut geeignet ist, und zeigen Sie einen Lösungsansatz, wie auch größere Firmen agile Methoden sinnvoll einsetzen können.

Fortsetzung nächste Seite!

Aufgabe 2 (Projektmanagement) [12 PUNKTE]

Betrachten Sie die folgende Tabelle zum Projektmanagement:

Arbeitspaket	Dauer (Tage)	abhängig von
A1	5	
A2	5	A1
A3	10	A2
A4	25	A1
A5	10	A1, A3
A6	10	
A7	5	A2, A3
A8	10	A4, A5, A6

a) Erstellen Sie ein Gantt-Diagramm, das die in der Tabelle angegebenen Abhängigkeiten berücksichtigt. Das Diagramm muss nicht maßstabsgetreu sein, jedoch jede Information aus der gegebenen Tabelle enthalten.

b) Wie lange dauert das Projekt mindestens?

c) Geben Sie alle kritischen Pfade an.

d) Bewerten Sie folgende Aussage eines Projektmanagers:
„Falls unser Projekt in Verzug gerät, bringen uns neue Programmierer auch nicht weiter."

Fortsetzung nächste Seite!

Aufgabe 3 (Use-Case-Diagramm) [21 PUNKTE]

Im Folgenden ist eine Systembeschreibung für einen „Fahrkartenautomat“ angegeben.

> Es gibt zwei Arten von Bahnfahrern: Gelegenheitsfahrer und Vielfahrer. Bei der Benutzung des Kartenautomaten kann sich grundsätzlich jeder die Hilfe anzeigen lassen und ein Beschwerdeformular ausfüllen.
> Gelegenheitsfahrer haben keine Benutzer-Id und können Einzelfahrkarten auswählen und anschließend kaufen. Damit ein Kaufvorgang erfolgreich abgeschlossen wird, muss ein entsprechender Geldbetrag eingezahlt werden. Dies geschieht entweder mit Bargeld oder per EC-Karte. Nach dem Kauf eines Tickets kann man sich dafür optional eine separate Quittung drucken lassen.
> Vielfahrer haben eine eindeutige Benutzer-Id mit Passwort, um sich am Automaten zu authentifizieren. Ein Vielfahrer kann sowohl eine Einzelfahrkarte als auch eine personalisierte Monatskarte erwerben. Sofern er eine Monatskarte besitzt (Information im System hinterlegt), kann er sich kostenfrei eine Ersatzfahrkarte ausstellen lassen, falls er seine Monatskarte verloren hat. Wenn die Authentifizierung oder ein Kaufvorgang fehlschlägt, soll eine entsprechende Fehlermeldung erscheinen.

a) Geben Sie die im Text erwähnten Akteure für das beschriebene System an.

b) Identifizieren Sie zwei weitere Stakeholder und nennen Sie dazu je zwei unterschiedliche Anwendungsfälle des Systems, in die diese involviert sind.

c) Geben Sie mindestens sechs verschiedene Anwendungsfälle für das beschriebene System an.

d) Erstellen Sie aus Ihren vorherigen Antworten ein Use-Case-Diagramm für das beschriebene System, in dem die Akteure und Anwendungsfälle inkl. möglicher Generalisierungen und Beziehungen eingetragen sind. Achten Sie insbesondere auf mögliche ⟨⟨include⟩⟩- und ⟨⟨extends⟩⟩- Beziehungen und Bedingungen für Anwendungsfälle.

Fortsetzung nächste Seite!

Aufgabe 4 (Objektorientierte Analyse) [16 PUNKTE]

Betrachten Sie das folgende Szenario:

> Entwickeln Sie für einen Kunden eine einheitliche Online-Plattform, in welcher mehrere Restaurants angebunden sind. Die Nutzer des Systems sollen Essen wie Pizzen, Burger oder Pasta bestellen und dabei aus einer Liste von verschiedenen Gerichten auswählen können. Die Plattform soll zusätzliche Optionen (z. B. Lieferung durch einen Lieferdienst oder direkte Abholung, inkl. Salat oder einer Flasche Wein) ermöglichen. Die Bestellung soll dann von der Plattform an den jeweiligen Gaststättenbetreiber gesendet werden. Die Besteller sollen zudem eine Bestätigung als Nachricht erhalten. Die jeweiligen Gerichte und Optionen haben unterschiedliche Preise, die dem Internetnutzer angezeigt werden müssen, bevor er diese auswählt. Der Endpreis muss vor der endgültigen Zahlung des Auftrags angezeigt werden. Kunden können (optional) einen Benutzeraccount anlegen und erhalten bei häufigen Bestellungen einen Rabatt.

a) Beschreiben Sie kurz ein Verfahren, wie Sie aus der Szenariobeschreibung mögliche Kandidaten für Klassen erhalten können.

b) Beschreiben Sie kurz ein Verfahren, um Vererbungshierarchien zu identifizieren.

c) Geben Sie **fünf** geeignete Klassen für das obige Szenario an. Nennen Sie dabei keine Klassen, welche durch Basisdatentypen wie Integer oder String abgedeckt werden können.

d) Nennen Sie **drei** Klassen für das obige Szenario, die direkt durch Basisdatentypen wie Integer oder String abgedeckt werden können.

e) Erstellen Sie ein Sequenzdiagramm für das gegebene System mit folgendem Anwendungsfall: Ein Nutzer bestellt zwei Pizzen und eine Flasche Wein. Beginnen Sie mit der „Auswahl des Gerichts“ bis hin zur „Bestätigung der Bestellung“ sowie „Lieferung an die Haustür“. Das Diagramm soll mindestens je einen Akteur für Benutzer (Browser), Applikation (Webserver) und Restaurant (Koch und Lieferdienst) vorsehen. Die Bezahlung selbst muss nicht modelliert werden.

Fortsetzung nächste Seite!

Aufgabe 5 (UML und Entwurfsmuster) [25 PUNKTE]

Wir betrachten Terme über die Rechenarten $op \in \{+, -, *, /\}$, die rekursiv definiert sind:

- Jedes Literal ist ein Term, z. B. „4“.
- Jedes Symbol ist ein Term, z. B. „x“.
- Ist t ein Term, so ist „(t)“ ein (geklammerter) Term.
- Sind t_1, t_2 Terme, so ist „t_1 op t_2“ ebenso ein Term.

Beispiele für gültige Terme sind also „$4 + 8$“, „$4 * x$“ oder „$4 + (8 * x)$“.

a) Welches Design-Pattern eignet sich hier am besten zur Modellierung dieses Sachverhalts?

b) Nennen Sie **drei** wesentliche Vorteile von Design-Pattern im Allgemeinen.

c) Modellieren Sie eine Klassenstruktur in UML, die diese rekursive Struktur von *Termen* abbildet. Sehen Sie mindestens einzelne Klassen für die *Addition* und *Multiplikation* vor, sowie weitere Klassen für *geklammerte Terme* und *Literale*, welche ganze Zahlen repräsentieren. Gehen Sie bei der Modellierung der Klassenstruktur davon aus, dass eine objektorientierte Programmiersprache wie Java zu benutzen ist.

d) Erstellen Sie ein Objektdiagramm, welches den Term $t := 4 + (8 * x) + (12 * y/(8 * x))$ entsprechend Ihres Klassendiagramms repräsentiert.

e) Überprüfen Sie, ob das Objektdiagramm für den in Teilaufgabe d) gegebenen Term eindeutig definiert ist. Begründen Sie Ihre Entscheidung.

f) Die gegebene Klassenstruktur soll mindestens folgende Operationen unterstützen:

- das Auswerten von Termen,
- das Ausgeben in einer leserlichen Form,
- das Auflisten aller verwendeten Symbole.

Welches Design-Pattern ist hierfür am besten geeignet?

g) Erweitern Sie Ihre Klassenstruktur um die entsprechenden Methoden, Klassen und Assoziationen, um die in Teilaufgabe f) genannten zusätzlichen Operationen gemäß dem von Ihnen genannten Design Pattern zu unterstützen.

Fortsetzung nächste Seite!

Aufgabe 6 (Softwarequalitätssicherung) [20 PUNKTE]

Gegeben sei folgendes Java-Programm, das mit der Absicht geschrieben wurde, den größten gemeinsamen Teiler zweier Zahlen zu berechnen:

```
1  /** Return the Greatest Common Divisor of two integer values. */
2  public int gcd(int a, int b) {
3    if (a < 0 || b < 0) {
4      return gcd(−b, a);
5    }
6    while (a != b) {
7      if (a > b) {
8        a = a − b;
9      } else {
10       b = b % a;
11     }
12   }
13   return a;
14 }
15
```

a) Bestimmen Sie eine möglichst kleine Menge an Testfällen für das gegebene Programm, um (dennoch) vollständige Zweigüberdeckung zu haben.

b) Welche Testfälle sind notwendig, um die Testfälle der Zweigüberdeckung so zu erweitern, dass eine 100 % Anweisungsüberdeckung erfüllt ist? Begründen Sie Ihre Entscheidung.

c) Beschreiben Sie zwei allgemeine Nachteile der Anweisungsüberdeckung.

d) Das gegebene Programm enthält (mindestens) zwei Fehler. Bestimmen Sie jeweils eine Eingabe, die fehlerhaftes Verhalten des Programms verursacht. Nennen Sie den Fault und den Failure, der bei der gewählten Eingabe vorliegt. Sie müssen das Programm (noch) nicht verbessern.

e) Geben Sie für die gefundenen zwei Fehler jeweils eine mögliche Verbesserung an. Es reicht eine textuelle Beschreibung, Code ist nicht notwendig.

f) Erläutern Sie, warum es im Allgemeinen hilfreich sein kann, bei der Fehlerbehebung in einem größeren Programm die Versionsgeschichte miteinzubeziehen.

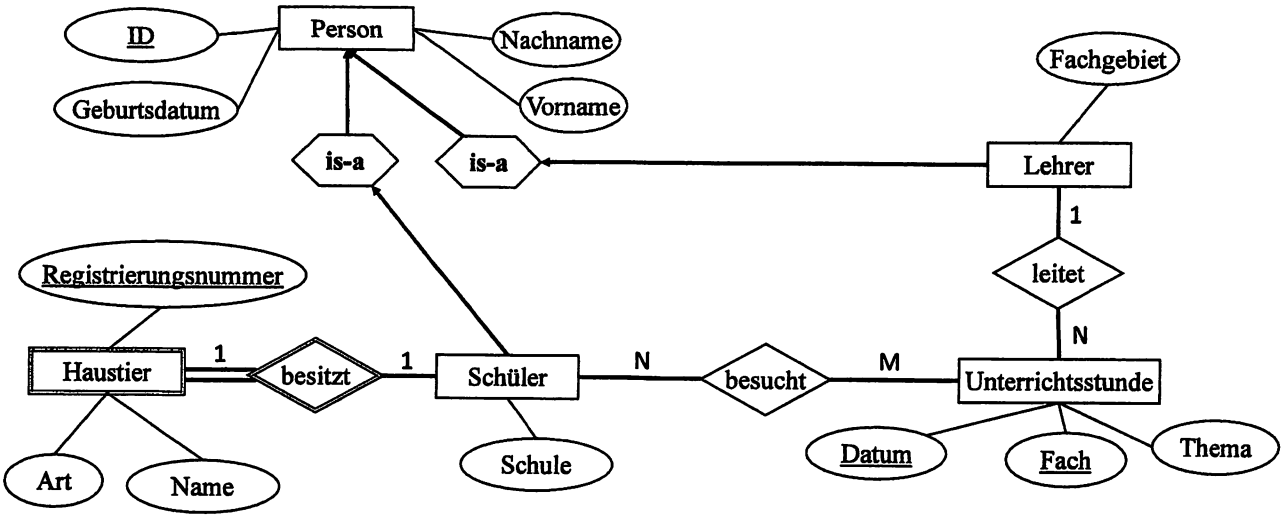
Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanksysteme

Aufgabe 1 (Entity-Relationship-Modellierung) [26 PUNKTE]

Hinweis: *Bei dieser Aufgabe wird Wissen über das erweiterte Entity-Relationship-Modell (bspw. schwache Entitätstypen, Vererbung) sowie die Verfeinerung eines relationalen Schemas vorausgesetzt.*

Gegeben ist das folgende ER-Modell, welches Beziehungen und Entitäten an einer Schule darstellt:



a) Beschreiben Sie dieses ER-Modell in eigenen Worten und gehen Sie dabei auf alle Entitäten, alle Relationen zwischen den Entitäten, sowie auch auf die Primärschlüssel genauer ein.

b) Überführen Sie dieses ER-Modell in ein verfeinertes relationales Schema. Wählen Sie geeignete Domänen für die Attribute und kennzeichnen Sie Schlüssel durch Unterstreichen, sowie Fremdschlüssel durch Überstreichen.

c) Warum kann die Beziehung besitzt zwischen den Entitätstypen Schüler und Haustier keine $N : M$-Beziehung sein?

d) Nehmen Sie an, dass das relationale Schema aus Teilaufgabe b) nun mit Hilfe von SQL-DDL-Anweisungen erstellt werden soll.

Wie kann folgender Constraint bei der Umsetzung berücksichtigt werden?
Die Schule der Schüler darf nur *DRW*, *ABC*, *RLW* oder *KWU* sein. Das Attribut darf darüber hinaus nicht NULL sein.

Fortsetzung nächste Seite!

Aufgabe 2 (SQL) [24 PUNKTE]

Gegeben ist der folgende Ausschnitt eines Schemas für die Verwaltung eines Restaurants.

Hinweis: Unterstrichene Attribute sind Primärschlüsselattribute, kursiv geschriebene Attribute sind Fremdschlüsselattribute.

Restaurant : {[
RestaurantID : INTEGER,
RestaurantName : VARCHAR(255),
StadtName : VARCHAR(255),
PLZ: INTEGER,
Straße : VARCHAR(255),
Hausnummer: INTEGER,
Kategorie : VARCHAR(255)
]}

Küche : {[
RestaurantID : INTEGER,
Art : VARCHAR(255),
KochPersonID : INTEGER
]}

Stadt : {[
StadtName : VARCHAR(255),
Land : VARCHAR(255)
]}

bevorzugt : {[
PersonID : INTEGER,
Art : VARCHAR(255)
]}

Person : {[
PersonID : INTEGER,
Name : VARCHAR(255),
Vorname : VARCHAR(255),
StadtName : VARCHAR(255),
PLZ: INTEGER,
Straße : VARCHAR(255),
Hausnummer: INTEGER
]}

Die Tabelle *Restaurant* beschreibt Restaurants eindeutig durch ihre ID. Zudem wird der Name, die Adresse des Restaurants und die (Sterne-)Kategorie gespeichert. *Küche* enthält u. a. Informationen zu der Art der Küche. Dabei kann ein Restaurant mehrere Arten anbieten, z. B. italienisch, deutsch, etc. In der Tabelle *Stadt* werden der Name der Stadt sowie das Land verwaltet, in dem die Stadt liegt. Wir gehen davon aus, dass eine Stadt eindeutig durch ihren Namen gekennzeichnet ist. *Person* beschreibt Personen mit Name, Vorname und Adresse. Personen werden eindeutig durch eine ID identifiziert. Die Tabelle *bevorzugt* gibt an, welche Person welche Art der Küche präferiert.

Bearbeiten Sie die folgenden Teilaufgaben:

a) Erläutern Sie kurz, warum das Attribut Art in Küche Teil des Primärschlüssels ist.

b) Schreiben Sie eine SQL-Anweisung, welche alle Städte findet, in denen man "deutsch" (Art der Küche) essen kann.

c) Schreiben Sie eine SQL-Abfrage, die alle Personen (Name und Vorname) liefert, die kein deutsches Essen bevorzugen. Verwenden Sie *keinen Join.*

d) Schreiben Sie eine SQL-Abfrage, die für jede Stadt (StadtName) und Person (PersonID) die Anzahl der Restaurants ermittelt, in denen diese Person bevorzugt essen gehen würde. Es sollen nur Städte ausgegeben werden, in denen es mindestens drei solche Restaurants gibt.

e) Schreiben Sie eine SQL-Abfrage, die die Namen aller Restaurants liefert, in denen sich die Personen mit den IDs 1 und 2 gemeinsam zum Essen verabreden können, und beide etwas zum Essen finden, das sie bevorzugen. Es sollen keine Duplikate ausgegeben werden.

Fortsetzung nächste Seite!

Aufgabe 3 (Relationale Algebra und Optimierung) [25 PUNKTE]

a) Betrachten Sie die Relation V. Sie enthält eine Spalte *Name* sowie ein dazugehörendes *Jahr*.

V	Name	Jahr
	A	2019
	A	2020
	B	2018
	B	2019
	B	2020
	C	2017
	C	2018
	C	2020

i. Gesucht ist eine Relation S, die das folgende Ergebnis von $V \div S$ berechnet ($\div$ ist die Division der relationalen Algebra):

$V \div S$	Name
	B

Welche der nachstehenden Ausprägungen für die Relation S liefert das gewünschte Ergebnis? Geben Sie eine Begründung an.

A)

S	Jahr
	2017
	2018
	2019
	2020

B)

S	Jahr
	2018
	2019
	2020

C)

S	Jahr
	2017
	2019
	2020

D)

weder A),

noch B),

noch C)

ii. Formulieren Sie die Divisions-Query aus Teilaufgabe i. in SQL.

Fortsetzung nächste Seite!

b) Gegeben sind die Tabellen $R(A, B)$ und $S(C, D)$ sowie die folgende View:

```
1 CREATE VIEW mv (A,C,D) AS
2     SELECT DISTINCT A, C, D
3     FROM R, S
4     WHERE B = D AND A <> 10;
```

Auf dieser View wird die folgende Query ausgeführt:

```
1 SELECT DISTINCT A
2 FROM mv
3 WHERE C > D;
```

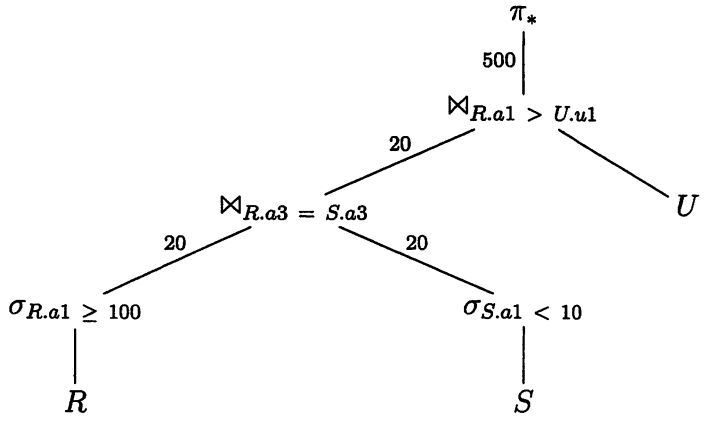
Konvertieren Sie die Query und die zugrundeliegenden View in einen Ausdruck der relationalen Algebra in Form eines Operatorbaums. Führen Sie anschließend eine relationale Optimierung durch. Beschreiben und begründen Sie dabei kurz jeden durchgeführten Schritt.

c) Gegeben sind die Relationen R, S und U sowie deren Kardinalitäten T_R, T_S und T_U:

R	(a1, a2, a3)	$T_R = 200$
S	(a1, a2, a3)	$T_S = 100$
U	(u1, u2)	$T_U = 50$

Bei der Ausführung des folgenden Query-Plans wurden die Kardinalitäten der Zwischenergebnisse mitgezählt und an den Kanten notiert.

Leiten Sie aus den Angaben im Ausführungsplan den Anteil der qualifizierten Tupel aller Prädikate her und geben Sie diese an.



Fortsetzung nächste Seite!

Aufgabe 4 (Entwurfstheorie) [20 PUNKTE]

Gegeben ist das folgende Relationenschema R in erster Normalform.

$$R : \{[A, B, C, D, E, F]\}$$

Für R gelte folgende Menge FD funktionaler Abhängigkeiten:
FD = {

$$\begin{aligned} AC &\longrightarrow DE \\ ACE &\longrightarrow B \\ E &\longrightarrow B \\ D &\longrightarrow F \\ AC &\longrightarrow F \\ AD &\longrightarrow F \end{aligned}$$

}

a) R mit FD hat genau einen Kandidatenschlüssel K. Bestimmen Sie diesen und begründen Sie Ihre Antwort.

b) Berechnen Sie Schritt für Schritt die Hülle X^+ von $X := \{K\}$.

c) Nennen Sie alle primen und nicht-primen Attribute.

d) Geben Sie die höchste Normalform an, in der sich die Relation befindet. Begründen Sie.

e) Gegeben ist die folgende Zerlegung von R:

R1 (A, C, D, E)

R2 (B, E)

R3 (D, F)

Weisen Sie nach, dass es sich um eine verlustfreie Zerlegung handelt.

Fortsetzung nächste Seite!

Aufgabe 5 (Transaktionen) [25 PUNKTE]

(a) Beschreiben Sie das Problem der *Abhängigkeit von nicht freigegebenen Änderungen* (Dirty Read) im Zusammenhang mit Mehrbenutzersynchronisation. Geben Sie hierfür ein geeignetes Beispiel in Form von zwei nebenläufigen Transaktionen an, welche bei unkontrolliertem Mehrbenutzerbetrieb zu einem Dirty Read führen.

(b) Gegeben ist die folgende Historie:

$w_2(a) \rightarrow r_1(c) \rightarrow r_4(a) \rightarrow r_2(c) \rightarrow r_3(a) \rightarrow r_4(d) \rightarrow r_3(c) \rightarrow r_3(b) \rightarrow w_4(b) \rightarrow r_1(d) \rightarrow c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow c_4$

i. Zeichnen Sie den Serialisierbarkeitsgraphen. Geben Sie außerdem bei den Pfeilen an, welche Ressource dabei zu dieser Konfliktsortierung geführt hat.

ii. Ist diese Historie serialisierbar? Geben Sie eine Begründung an.

iii. Geben Sie alle möglichen äquivalenten seriellen Historien an. *Hinweis: Es genügt, wenn Sie die Transaktionen in der folgenden Form angeben: z.B.* $T_xT_yT_z$, $T_xT_zT_y$, ...

(c) Gegeben ist der folgende Schedule für die Transaktionen T_1, T_2 und T_3:

T_1	T_2	T_3
	BOT	
	read(F, x_2)	
	$x_2 := x_2 - 10$	
BOT		
read(F, x_1)		
	write(x_2, F)	
	commit	
		BOT
		read(F, x_3)
		$x_3 := x_3 * 2$
$x_1 := x_1 + x_1$		
write(x_1, F)		
commit		
		write(x_3, F)
		abort

Berechnen Sie den Wert von F nachdem alle Transaktionen mit ihren Aktionen fertig sind. Gehen Sie von einem Anfangswert von 20 für F aus. Erläutern Sie kurz, wie das Ergebnis zustande kommt.