

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

**Kennzahl:** \_\_\_\_\_

**Kennwort:** \_\_\_\_\_

**Arbeitsplatz-Nr.:** \_\_\_\_\_

**Frühjahr  
2014**

**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:** 14

---

**Bitte wenden!**

## Thema Nr. 1

### Teilaufgabe 1:

#### 1. Modellierung

Der VDS (Verein deutscher Schwimmbäder) möchte Informationen über seine bestehenden Einrichtungen in Verbindung mit dessen Besucherzahlen festhalten und digitalisieren. Dazu soll eine Datenbank in zwei Schritten erstellt werden. Der erste soll folgende Informationen enthalten:

- Personen werden spezifiziert durch ihren Vor- sowie Nachnamen und einer Ausweisnummer.
- Ein Schwimmbad besitzt einen Namen und eine Stadt, in der es sich befindet.
- Personen besuchen Schwimmbäder.
- Für Schwimmbecken wird ein Name, die Länge, die Breite sowie die Tiefe gespeichert. Ein Schwimmbecken befindet sich in genau einem Schwimmbad und kann nur durch dieses eindeutig identifiziert werden.
- Personen können in einem Schwimmbecken schwimmen, wobei dazu ein Schwimmstil festgehalten werden soll.

Im zweiten Schritt der Datenbank wird spezieller auf die Personen im Schwimmbad eingegangen:

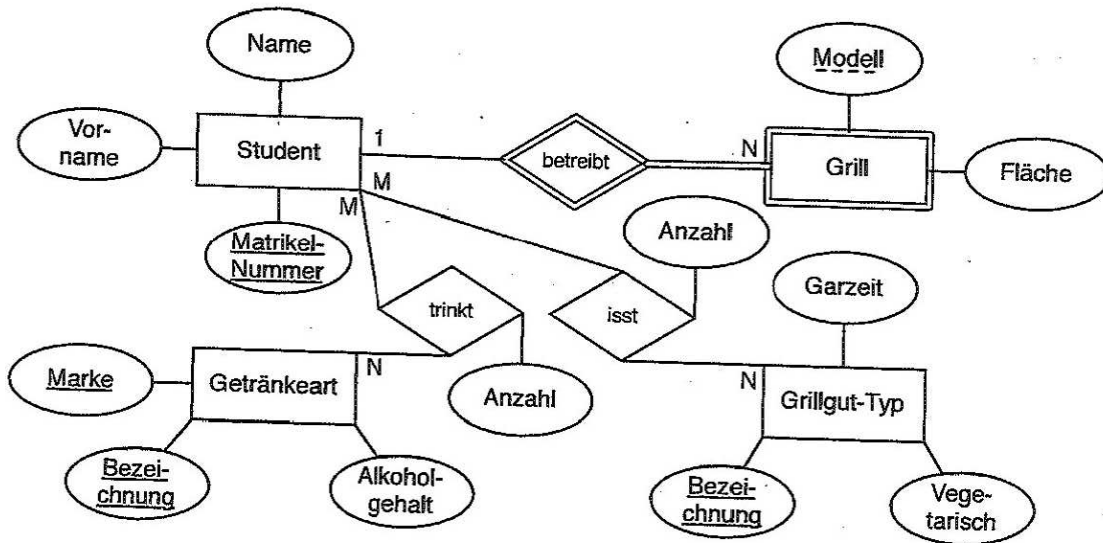
- Im Bezug auf die Domäne unterteilen sich Personen in Personal des Schwimmbads und Besucher. Angestellte des Schwimmbads können dieses aber auch außerhalb Ihrer Arbeitszeiten besuchen. Für das Personal wird eine Personalnummer, für die Besucher eine Besuchernummer gespeichert.
- Es werden Schwimmkurse angeboten. Zu diesen wird eine Kursnummer sowie dessen Gebiet festgehalten.
- Ein Angestellter kann eine beliebige Anzahl an Kursen leiten, wobei jeder Kurs eine beliebige Anzahl an Besuchern als Teilnehmer fassen kann.

**Fortsetzung nächste Seite!**

- a) Entwerfen Sie für das beschriebene Szenario des **ersten** Schritts der Datenbank ein ER-Modell! Bestimmen Sie hierzu:
- die Entity-Typen, die Relationship-Typen und jeweils deren Attribute,
  - die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen, und
  - die Kardinalitäten der Relationship-Typen.
- b) Übernehmen Sie den Entity-Typ für eine Person aus Aufgabenteil a) und entwerfen Sie ausgehend davon ein weiteres ER-Modell für den **zweiten** Teil der Datenbank! Bestimmen Sie dazu ebenfalls dieselben Kriterien wie in Aufgabenteil a)!
- c) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell! Geben Sie hierfür die verallgemeinerten Relationenschemata an! Achten Sie dabei insbesondere darauf, dass die Relationenschemata keine redundanten Attribute enthalten!

## 2. SQL-Anfragen

Gegeben sei folgendes ER-Modell, welches eine Datenbank für Statistiken über Grillabende beschreibt. Darin sind Studenten, die Grills betreiben, enthalten. Zu jedem Studenten soll die Anzahl an Speisen und Getränken erfasst werden, welche sie zu sich genommen haben:



Gehen Sie von folgendem dazugehörigen relationalem Schema aus:

Student{[Matrikel-Nummer, Name, Vorname]}  
 Grill{[Modell, Matrikel-Nummer, Fläche]}  
 Getränk{[Marke, Bezeichnung, Alkoholgehalt]}  
 Grillgut-Typ{[Bezeichnung, Vegetarisch, Garzeit]}  
 Trinkt{[Getränk-Mark, Getränk-Bezeichnung, Matrikel-Nummer, Anzahl]}  
 Isst{[Grillgut-Typ-Bezeichnung, Matrikel-Nummer, Anzahl]}

Formulieren Sie folgende Anfragen in SQL. Achten Sie, falls nötig, auf duplikatfreie Ausgaben:

- Geben Sie die Garzeit des Grillgut-Typs mit der Bezeichnung „Schweinenacken“ aus!
- Geben Sie die Namen und Vornamen aller Grillmeister aus! *Hinweis:* Als Grillmeister gelten Studierende, die mindestens einen Grill betreiben.
- Geben Sie alle Attribute von Studierenden aus, die mindestens ein Getränk mit mehr als 5 % Alkohol getrunken haben!
- Geben Sie alle Attribute von Studierenden aus, die mindestens zwei alkoholhaltige Getränke konsumiert haben!

Fortsetzung nächste Seite!

## 3. Normalformen

Gegeben sei folgendes relationales Schema  $R$ , dessen Attribute nur atomare Attributwerte besitzen.

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

Es gelte die folgende Menge  $FD$  von funktionalen Abhängigkeiten:

$FD = \{$

$$A \rightarrow B,$$

$$CD \rightarrow EB,$$

$$C \rightarrow DA,$$

$$CF \rightarrow FE,$$

$$F \rightarrow GHCA$$

$\}$

- Nennen Sie die drei Änderungsanomalien, die auftreten können, wenn ein Datenbank-Schema nicht der Boyce-Codd-Normalform entspricht, und beschreiben Sie eine näher!
- Definieren Sie die Begriffe Superschlüssel, Schlüsselkandidat und Primarschlüssel. Wie unterscheiden sich diese voneinander?
- Finden Sie den/die Schlüsselkandidaten!
- Prüfen Sie, in welcher Normalform  $R$  vorliegt!
- Berechnen Sie die kanonische Überdeckung  $FD_c$  von  $FD$ !  
*Hinweis:* Es genügt, wenn Sie für jeden der vier Einzelschritte die Menge der funktionalen Abhängigkeiten als Zwischenergebnis angeben.
- Bestimmen Sie eine Zerlegung der Relation  $R$  in dritter Normalform. Verwenden Sie dafür den Synthesealgorithmus!

*Hinweis:* Sie können dazu die kanonische Überdeckung  $FD_c$  verwenden, die Sie in der vorherigen Teilaufgabe berechnet haben.

**Fortsetzung nächste Seite!**

**Teilaufgabe 2:**

Es soll ein Framework für Zweipersonenspiele, wie z.B. Mühle, Schach, Reversi, Tic-Tac-Toe etc. erstellt werden, welches es erleichtert, solche Spiele als Handyanwendung schnell und kostengünstig zu lancieren.

In all diesen Spielen sind zwei Spieler beteiligt, die abwechselnd ziehen. Das Framework soll aufgrund vorher festgelegter Regeln überprüfen, ob ein Zug erlaubt ist und das Ergebnis des Zuges auf dem Bildschirm anzeigen. Es muss die Möglichkeit geben, Spielregeln, die Art der Züge, die Geometrie und das konkrete Aussehen des Spielfeldes festzulegen. Außerdem sollen bisher stattgefundenen Partien abgespeichert werden können und eine Bestenliste verwaltet werden.

1. Skizzieren Sie einen objektorientierten Entwurf für diese Aufgabenstellung in Form eines UML Klassendiagramms. Ihr Diagramm sollte nicht weniger als sieben Klassen haben und neben den Beziehungen der Klassen untereinander auch deren wichtigste Methoden und Attribute beinhalten. Sie können sich zunächst an einem bestimmten Spiel orientieren, müssen aber die allgemeine Einsetzbarkeit sicherstellen.
2. Erstellen Sie ein Ablaufdiagramm, aus dem hervorgeht, welcher Spieler am Zug ist, ob ein illegaler Zug durchgeführt wurde und ob das Spiel zuende ist.
3. Erläutern Sie detailliert zwei Entwurfsmuster Ihrer Wahl. Geben Sie insbesondere entsprechende Klassendiagramme an und skizzieren Sie die jeweiligen Einsatzmöglichkeiten. Es wird empfohlen, solche Muster zu wählen, die in Ihrem Entwurf auch eingesetzt wurden.

## Thema Nr. 2

### Teilaufgabe 1:

Aufgabe 1: E/R-Modellierung, SQL-DDL

Für die Verwaltung von Museen soll folgendes System entworfen werden:

- Die zentrale Komponente ist das Museum mit eindeutigen Namen.
- Ein Museum besteht aus mindestens einem Themenbereich, welcher ebenfalls einen eindeutigen Namen besitzt.
- Ein Themenbereich erstreckt sich über mindestens einen Raum, der einen Namen hat und über eine eindeutige Raumnummer identifiziert werden kann. Ein Raum gehört zu genau einem Themenbereich.
- In einem Raum befindet sich mindestens ein Kunstwerk. Ein Kunstwerk gehört zum Eigentum genau eines Museums, welches aber (im Fall einer Leihgabe) nicht zwingend das Museum sein muss, in dem es ausgestellt ist. Ein Museum kann Eigentümer vieler Kunstwerke sein.
- Ein Kunstwerk wurde von genau einem Künstler erstellt und gehört einer speziellen Kunstepoche an. Jedes Kunstwerk hat einen eindeutigen Titel sowie einen Untertitel und ein Entstehungsjahr.
- Eine Kunstepoche wird durch einen eindeutigen Namen und einen Zeitraum (Start- und Endjahr) beschrieben.
- Ein Künstler wird durch seinen Namen und Vornamen identifiziert und hat außerdem ein Geburtsdatum und gegebenenfalls ein Sterbedatum. Von Künstlern kann man sagen, dass sie in ihrem Leben viele Kunstwerke erschaffen und in mindestens einer Kunstepoche gewirkt haben.
- Manche Künstler sind Mitglied in einer Künstlervereinigung mit eindeutigen Namen und einem Gründungsdatum.

1. Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank!
2. Setzen Sie das gegebene E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Geben Sie die resultierenden Relationenschemata in folgender Schreibweise an:

Relation (Attribut1, Attribut2, ..., AttributN)

Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen! Achten Sie auf eine möglichst kompakte Modellierung der Relationships! Geben Sie außerdem erforderliche Fremdschlüsselbeziehungen an!

3. Geben Sie die Anweisungen in SQL-DDL an, die notwendig sind, um die Relationen für ein **Kunstwerk** und eine **Kunstepoche** aus Teilaufgabe (2) in einer relationalen Datenbank zu erzeugen! Kennzeichnen Sie dabei die Primär- und Fremdschlüssel der Relationen. Geben Sie sinnvolle Integritätsbedingungen für Zahlen und Daten an und achten Sie auf die sinnvolle Behandlung möglicher leerer Ausprägungen!

**Fortsetzung nächste Seite!**

## Aufgabe 2: Relationale Algebra und SQL

Gegeben sei das folgende relationale Schema mitsamt Beispieldaten für eine Datenbank für Bierbrauereien und ihre Produkte. Die Primärschlüssel-Attribute sind jeweils unterstrichen, Fremdschlüssel sind überstrichen.

"Brauerei":

<u>ID</u>	Name	<u>Sitz</u>	Gründungsjahr
B1	Augustiner	S4	1328
B2	Beck's	S3	1873
B3	Paulaner	S4	1634
B4	Krombacher	S5	1803
B5	Früh	S1	1904
B6	Gaffel	S1	1908
B7	Hoepfner	S2	1798
B8	Spaten	S4	1397

"Sitz":

<u>ID</u>	Stadt	Bundesland
S1	Köln	Nordrhein-Westfalen
S2	Karlsruhe	Baden-Württemberg
S3	Bremen	Bremen
S4	München	Bayern
S5	Krombach	Nordrhein-Westfalen

"Biersorte":

<u>ID</u>	Bezeichnung	Hefetyp
T1	Pils	untergärig
T2	Lager	untergärig
T3	Weizenbier	obergärig
T4	Helles	untergärig
T5	Starkbier	ober- und untergärig
T6	Export	untergärig
T7	Kölsch	obergärig
T8	Altbier	obergärig

"Bier":

<u>ID</u>	<u>Brauerei</u>	<u>Biersorte</u>	Alkoholgehalt
A01	B1	T4	5,2
A02	B1	T6	5,6
A03	B2	T1	4,9
A04	B3	T3	5,5
A05	B3	T4	4,9
A06	B4	T1	4,8
A07	B5	T7	4,8
A08	B6	T7	4,8
A09	B7	T1	4,7
A10	B8	T4	5,2

1. Formulieren Sie die folgenden Anfragen auf das gegebene Schema in relationaler Algebra:

- Finden Sie die Namen aller Brauereien, die vor dem Jahr 1800 gegründet wurden.
- Finden Sie die Bezeichnungen aller Biersorten mit einem Alkoholgehalt von 4,8% oder 4,9%.
- Finden Sie die Städte und zugehörigen Bundesländer, in denen Brauereien ihren Sitz haben, die im 19. Jahrhundert gegründet wurden.
- Finden Sie die Bezeichnungen aller Biersorten, die von der Brauerei 'Paulaner' hergestellt werden.

2. Geben Sie das Ergebnis (bezüglich der Beispieldaten) der folgenden Ausdrücke der relationalen Algebra als Tabellen an:

- $\pi_{Name, Bundesland}(\text{Brauerei} \bowtie_{\text{Brauerei.Sitz}=\text{Sitz.ID}} \text{Sitz})$
- $\pi_{Gründungsjahr, Hefetyp}(\text{Brauerei} \bowtie_{\text{Brauerei.ID}=\text{Bier.Brauerei}} \text{Bier} \bowtie_{\text{Bier.Biersorte}=\text{Biersorte.ID}} \sigma_{\text{Hefetyp} = \text{'obergärig'}}(\text{Biersorte}))$

Fortsetzung nächste Seite!



3. Formulieren Sie die folgenden Anfragen in SQL:

- a) Finden Sie die Namen aller Brauereien, die mehr als eine Biersorte herstellen!
- b) Geben Sie die Bezeichnung einer jeden Biersorte zusammen mit ihrem durchschnittlichen Alkoholgehalt an!
- c) Geben Sie die Stadt aus, in der die älteste Brauerei ihren Sitz hat!
- d) Finden Sie alle Bundesländer mit Brauereien, die nur untergärige Biersorten herstellen!

4. Wie sieht die Ergebnisrelation zu folgenden Anfragen auf den Beispieldaten aus?

- a) 

```
select Hefetyp, count(Bezeichnung) as Anzahl
from Biersorte
group by Hefetyp
having count(Bezeichnung) > 1;
```
- b) 

```
Select Bezeichnung
from Biersorte
where ID not in
(select Biersorte
 from Bier join Brauerei on Bier.Brauerei = Brauerei.ID
      join Sitz on Brauerei.Sitz = Sitz.ID
 where Bundesland = 'Bayern');
```

**Fortsetzung nächste Seite!**

## Aufgabe 3: Entwurfstheorie

Gegeben sei das folgende relationale Schema mitsamt Beispieldaten für eine Flugdatenbank. Die Primärschlüsselattribute sind unterstrichen.

Relation "Flüge":

<u>FNr</u>	<u>Datum</u>	VonS	VonL	NachS	NachL	Gesellschaft	Sitz	Flugzeug	Typ	Plätze
LH1432	03.04.11	München	GER	Frankfurt	GER	Lufthansa	Köln	Boeing	747	452
UA732	05.04.12	Chicago	USA	Brisbane	AUS	United	Chicago	Embraer	ERJ-195	108
UA735	05.04.12	Frankfurt	GER	Chicago	USA	United	Chicago	Airbus	320	152
EK935	01.02.11	Dubai	UAE	Montreal	CAN	Emirates	Dubai	Airbus	380	526
UA735	05.04.09	Frankfurt	GER	Chicago	USA	United	Chicago	Airbus	320	152

Gegeben seien ebenso die folgenden funktionalen Abhängigkeiten:

FD1:  $\text{FNr} \rightarrow \text{VonS}, \text{VonL}, \text{NachS}, \text{NachL}, \text{Gesellschaft}, \text{Sitz}, \text{Flugzeug}, \text{Typ}, \text{Plätze}$

FD2:  $\text{VonS} \rightarrow \text{VonL}$

FD3:  $\text{NachS} \rightarrow \text{NachL}$

FD4:  $\text{Gesellschaft} \rightarrow \text{Sitz}$

FD5:  $\text{Flugzeug}, \text{Typ} \rightarrow \text{Plätze}$

1. Beschreiben Sie kurz, welche Redundanzen in der Datenbank vorhanden sind und welche Anomalien auftreten können. Geben Sie ein Beispiel für jede Anomalie an.
2. Welchen Normalformen genügt das angegebene Relationenschema? Begründen Sie Ihre Entscheidung.
3. Überführen Sie das obige Relationenschema in die dritte Normalform und geben Sie die mit obigen Daten gefüllten Relationen an.
4. Erläutern Sie kurz, welchen Nachteil Normalisierung allgemein für die Anfragebearbeitung haben kann.

Fortsetzung nächste Seite!

**Teilaufgabe 2:****Aufgabe 1 (Objektorientierung)**

In dieser Aufgabe werden Sie Datentypen für die Verwaltung eines Parkhauses mit Hilfe objektorientierter Methoden definieren. Bearbeiten Sie die folgenden Teilaufgaben in einer objektorientierten Programmiersprache Ihrer Wahl (geben Sie diese an)! Solange nicht anders definiert, sollen Eigenschaften und Methoden **uneingeschränkt sichtbar** sein.

(a) Erzeugen Sie eine **Klasse Fahrzeug**, deren Instanzen folgende Eigenschaften besitzen (wählen Sie geeignete Typen):

- Ein amtliches Kennzeichen (Buchstaben- und Zahlenkombination).
- Die Dimensionen des Fahrzeugs (Länge, Breite, Höhe) in Metern.
- Das Datum der Erstzulassung. Definieren Sie hierfür entweder einen eigenen Datentyp oder machen Sie Gebrauch von der Standardbibliothek Ihrer gewählten Programmiersprache.

Die Eigenschaften sollen für **Unterklassen nicht sichtbar** sein. Schreiben Sie außerdem einen **Konstruktor**, der eine Instanz erzeugt und die Eigenschaften setzt!

(b) Schreiben Sie eine **Klasse Parkplatz**, in der ebenfalls Eigenschaften für die Dimension (Länge, Breite, Höhe) in Metern vorgesehen sind! Die Eigenschaften sollen ebenfalls im **Konstruktor** initialisiert werden können. Außerdem soll eine **Objektmethode** hinzugefügt werden, die prüft, ob ein gegebenes Fahrzeug in den Parkplatz passt.

(c) Ein **Interface Parkhaus** soll **Objektmethoden** für folgende Anwendungsfälle **deklarieren**:

- (i) Alle freien Parkplätze sollen (z.B. als **Array** oder als Instanz einer in der Standardbibliothek Ihrer verwendeten Sprache definierten **Kollektionsklasse**) zurückgegeben werden.
- (ii) Der erste freie Parkplatz, der zu einem gegebenen Fahrzeug passt, soll zurückgegeben werden.
- (iii) Ein gegebener Parkplatz soll für ein gegebenes Fahrzeug reserviert werden.

Deklarieren Sie das Interface und geben Sie geeignete Signaturen für die **Objektmethoden** an!

(d) Schreiben Sie eine **abstrakte Klasse**, die das Interface **Parkhaus** **partiell** implementiert! Geben Sie außerdem eine geeignete **Implementierung** für die Objektmethode (c.ii) unter Verwendung der existierenden Objektmethoden aus (b) und (c.i) an!

**Fortsetzung nächste Seite!**

**Aufgabe 2 (Klassendiagramm)**

Erstellen Sie zu der folgenden Beschreibung eines Systems zur Organisation eines Supermarktes ein UML-Klassendiagramm, das Attribute und Assoziationen mit Kardinalitäten, Rollennamen und Namen, sowie Methoden enthält! Setzen Sie dabei das Konzept der Vererbung sinnvoll ein!

- Ein Supermarkt hat einen Namen.
- Er wirbt und kann eine Inventur durchführen. Bei der Werbung werden bestimmte Waren beworben, die im Vorfeld festgelegt werden. Bei einer Inventur wird zurückgegeben, ob diese erfolgreich war.
- Der Supermarkt besteht aus beliebig vielen Regalen. Ein Regal gehört zu genau einem Supermarkt.
- Ein Regal hat einen Namen und eine eindeutige Nummer.
- Innerhalb eines Supermarktes befinden sich beliebig viele Waren. Jede Ware befindet sich in genau einem Supermarkt.
- Eine Ware hat einen Namen und einen Preis.
- Eine Ware ist einem Regal zugeordnet. Ein Regal beinhaltet beliebig viele, jedoch mindestens eine Ware.
- Im Kaufhaus arbeiten Verkäufer. Diese haben einen Namen und eine Personalnummer.
- Verkäufer werden zudem in Regalbetreuer und Kassierer unterschieden.
- Ein Verkäufer ist in mindestens einem Kaufhaus beschäftigt. Ein Kaufhaus beschäftigt beliebig viele Verkäufer.
- Ein Kassierer kennt den von ihm gemachten Umsatz.
- Ein Regalbetreuer hat einen Zuständigkeitsbereich, der als Zeichenkette repräsentiert wird.
- Beliebige viele Kunden kaufen in beliebig vielen Supermärkten ein und werden von Verkäufern beraten.
- Ein Verkäufer kann Auskunft über den Preis einer Ware geben, wenn man ihm die Ware übergibt.
- Ein Kunde kann einen Supermarkt mit gut oder schlecht bewerten. (Methode!)

**Fortsetzung nächste Seite!**

**Aufgabe 3 (Projektmanagement)**

Die Abbildung 1 zeigt ein CPM-Netzwerk mit topologisch sortierten Knoten. Die Vorgänge tragen keine Namen. Gestrichelte Linien zwischen Ereignissen stellen Scheinvorgänge (auch Pseudoaktivitäten) mit einer Dauer von 0 dar.

- (a) Welche Scheinvorgänge könnten aus dem Netzwerk entfernt werden, ohne dass Informationen verloren gehen? (mit Begründung)
- (b) Berechnen Sie für jedes Ereignis den frühesten Termin, wobei angenommen wird, dass das Projekt zum Zeitpunkt 0 startet! Geben Sie den frühestmöglichen Endtermin des Projekts an!
- (c) Berechnen Sie für jedes Ereignis auch die spätesten Zeiten, indem Sie für das letzte Ereignis den frühesten Termin als spätesten Termin ansetzen!
- (d) Geben Sie nun die Pufferzeiten der Ereignisse an!
- (e) Wie verläuft der kritische Pfad durch das Netzwerk?
- (f) Was bedeutet es für die Gesamtprojektdauer, wenn sich ein Vorgang auf dem kritischen Pfad um zwei Zeiteinheiten verzögert?

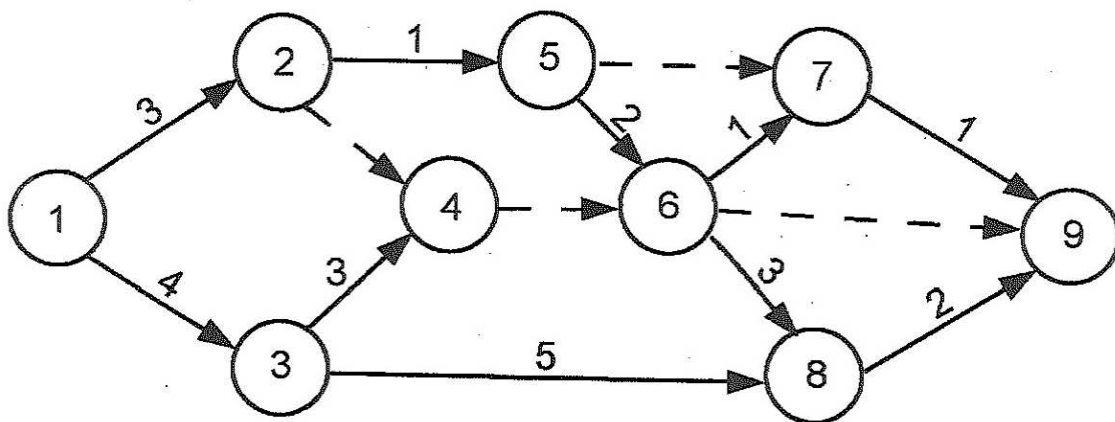


Abbildung 1: CPM-Netzwerk

**Fortsetzung nächste Seite!**

**Aufgabe 4****a) Anwendungsfalldiagramm**

Gegeben ist die folgende textuelle Beschreibung für einen **Anwendungsfall** aus dem Szenario *App-Entwicklung für eine Spedition*.

**Name** Übergabe der Ware an den Kunden

**Akteure** Speditionsfahrer, Kunde

**Vorbedingung** Die Tagesplanung der Spedition ist abgeschlossen.

**Ablauf** Dieser Anwendungsfall beschreibt einen normalen Ablauf für eine Auslieferung. Der Fahrer holt sich vorab die Information, welche Aufgaben anstehen und lässt seine Route planen. Er kündigt telefonisch die Lieferung an und stimmt sich mit dem Kunden ab. Bei Anlieferung übergibt der Fahrer die Fracht und lässt sich den Erhalt quittieren. Die erfolgreiche Auslieferung wird vom Fahrer an den zentralen Server zurückgemeldet.

**Nachbedingung** Der Auftrag ist erledigt.

**Priorität** hoch

Modellieren Sie den Anwendungsfall in einem **UML-Anwendungsfalldiagramm**!

**b) Aktivitätsdiagramm**

Erstellen Sie ein **UML-Aktivitätsdiagramm**, welches den Ablauf des Anwendungsfalls „Übergabe der Ware an den Kunden“ aus der vorangegangenen Aufgabe modelliert! Sehen Sie für jeden Akteur sowie für die Speditionszentrale eine **Partition** vor! Simulieren Sie dabei Telefonat, Frachtübergabe und Quittierung als **Aktionsknoten** und überlegen Sie sich, welche zusätzlichen **Objektknoten** und **-flüsse** notwendig sind!