

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	Frühjahr 2020	66116
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

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: 19

Bitte wenden!

Thema Nr. 1
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1 (Zustandsdiagramm)

[20 PUNKTE]

a) Basisfunktion [12 Punkte]

Erstellen Sie ein Zustandsdiagramm für einen Bankautomat, welcher den im Folgenden beschriebenen Authentifizierungsvorgang von Bankkunden realisiert. Modellieren Sie dazu soweit nötig sowohl Zustände und Transitionsbedingungen als auch die Aktionen der Zustände.

Der Bankautomat startet im Grundzustand und wartet auf das Einlegen einer Bankkarte. Wird eine Karte eingelegt, wird diese eingezogen und der Automat startet die Überprüfung der Bankkarte. Ist die Karte ungültig, wird die Karte ausgeworfen und der Automat wechselt in den Grundzustand. Ist die Karte gültig, kann die vierstellige PIN eingelesen werden. Nach der Bestätigung der Eingabe wird diese überprüft. Ist die PIN gültig, so stoppt der Automat und zeigt eine erfolgreiche Authentifizierung an. Ist die PIN ungültig, zeigt der Automat einen Fehler an und erlaubt eine erneute Eingabe der PIN.

b) Erweiterung [8 Punkte]

Der Bankautomaten aus Aufgabe a) soll nun so verändert werden, dass ein Bankkunde nach dem ersten fehlerhaften Eingeben der PIN die PIN erneut eingeben muss. Bei erneuter Falscheingabe, wird eine dritte Eingabe möglich. Bei der dritten Falscheingabe der PIN wird die Karte vom Automaten eingezogen und der Automat geht wieder in den Ausgangszustand über.

Hinweis: Für diese Aufgabe dürfen Sie Ihr Zustandsdiagramm aus a) weiter verwenden, wenn Sie eindeutig, z. B. durch den Einsatz von Farben kennzeichnen, was nur zur Aufgabe a) gehört und was Abänderungen des Zustandsdiagramms aus a) sind. Sie können, falls Sie einen neuen Automaten zeichnen, Zustände und Übergänge, die inhaltsgleich zur Lösung des Aufgabenteils a) sind mit einem "W" markieren, statt sie zu beschriften. In diesem Fall wird der Text aus der Lösung zu Aufgabenteil a) an dieser Stelle wiederholt gedacht.

Aufgabe 2 (UML-Klassendiagramm)

[34 PUNKTE]

Modellieren Sie mit Hilfe eines UML-Klassendiagramms das wie folgend spezifizierte Kurssystem. In dieser Aufgabe soll das Vorlesungs- und Klausurverwaltungssystem einer Universität modelliert werden.

Als Teilnehmer gibt es Dozenten und Studenten. Für Studenten soll eine Matrikelnummer und eine E-Mail-Adresse gespeichert werden, für Dozenten eine Mitarbeiternummer. Zusätzlich sind sowohl Dozenten als auch Studenten Menschen und für jeden Menschen wird Vorname, Nachname und Geburtsdatum gespeichert.

Fortsetzung nächste Seite!

Ein Dozent kann eine Klausur mit einem bestimmten Datum erstellen. Jeder Student kann sich für beliebig viele Klausuren anmelden. Nachdem der Dozent eine Klausur benotet hat, bekommt jeder teilnehmende Student eine Note für diese Klausur.

Vorlesungen haben einen Namen, eine Beschreibung und können entweder verpflichtend oder freiwillig sein. Zu jeder Vorlesung kann es beliebig viele Klausuren geben. Zusätzlich kann eine Vorlesung beliebig viele andere Vorlesungen als Voraussetzungen haben.

Hinweis: Achten Sie auf das Vorhandensein von beschrifteten Assoziationen, Attributen und Methoden und nutzen Sie Vererbungsstrukturen, wenn dies angemessen erscheint. Auf Sichtbarkeiten und Parameter kann verzichtet werden.

Aufgabe 3 (Projektmanagement)

[15 PUNKTE]

Ein Team aus Softwareentwicklern soll ein Projekt umsetzen, das in 7 Arbeitspakete unterteilt ist. Die Dauer der Arbeitspakete und ihre Abhängigkeiten können Sie aus der folgenden Tabelle entnehmen. Es kann pro Arbeitspaket nur ein Entwickler arbeiten.

Name	Dauer in Wochen	Abhängig von
A1	3	-
A2	6	-
A3	3	A2
A4	4	A1,A2
A5	6	A4
A6	8	A3
A7	4	-

Tabelle 1: Arbeitspakete

a) Gantt-Diagramm [11 Punkte]

Zeichnen Sie ein Gantt-Diagramm (Balkenplan), das eine kürzestmögliche Projektabwicklung beinhaltet.

b) Anwendung [4 Punkte]

1. Geben Sie den Kritischen Pfad und seine Dauer an.
2. Wie viele Entwickler sind maximal gleichzeitig im Projekt beschäftigt?
3. Um wie viele Wochen würde sich das Projekt verlängern, wenn ein Entwickler weniger zur Verfügung stünde?

Fortsetzung nächste Seite!

Aufgabe 4 (Entwicklungsprozesse)**[10 PUNKTE]**

Wählen Sie zwei Prozessmodelle.

- Nennen und erläutern Sie kurz (1-3 Sätze) zu jedem dieser Modelle drei wichtige Merkmale.
- Geben Sie je zwei Beispiele für Projekteigenschaften an, bei denen eines der gewählten Modelle Vorteile gegenüber dem anderen hat.

Aufgabe 5 (Implementierung)**[21 PUNKTE]**

Gegeben Sei das folgende Stück Code:

```
// my implementation
class Example
{
    /**
     * The helper
     */
    static int helper(int arr[], int x, int HIGH)
    {
        int sec = arr[HIGH];
        int i = (x-1);
        for (int j=x; j<HIGH; j++)
        {
            if (arr[j] <= sec)
            {
                int temp = arr[++i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
            int temp = arr[i+1];
            arr[i+1] = arr[HIGH];
            arr[HIGH] = temp;
            return i+1;
        }
    }
    /**
     * The core
     */
    static void core(int arr[], int x, int HIGH)
    {
        if (x < HIGH)
        {
            int pi = helper(arr, x, HIGH);
            core(arr, x, pi-1);
            core(arr, pi+1, HIGH);
        }
    }
    /**
     * Performs the task.
     */
    public static void main(String args[])
    {
        int arr[] = {10, 7, 8, 9, 1, 5};
        int n = arr.length;
        core(arr, 0, n-1);
        printArray(arr);
    }
}
```

- Funktion [1 Punkt]

Benennen Sie, welcher bekannte Algorithmus sich hinter diesem Stück Code versteckt.

Fortsetzung nächste Seite!

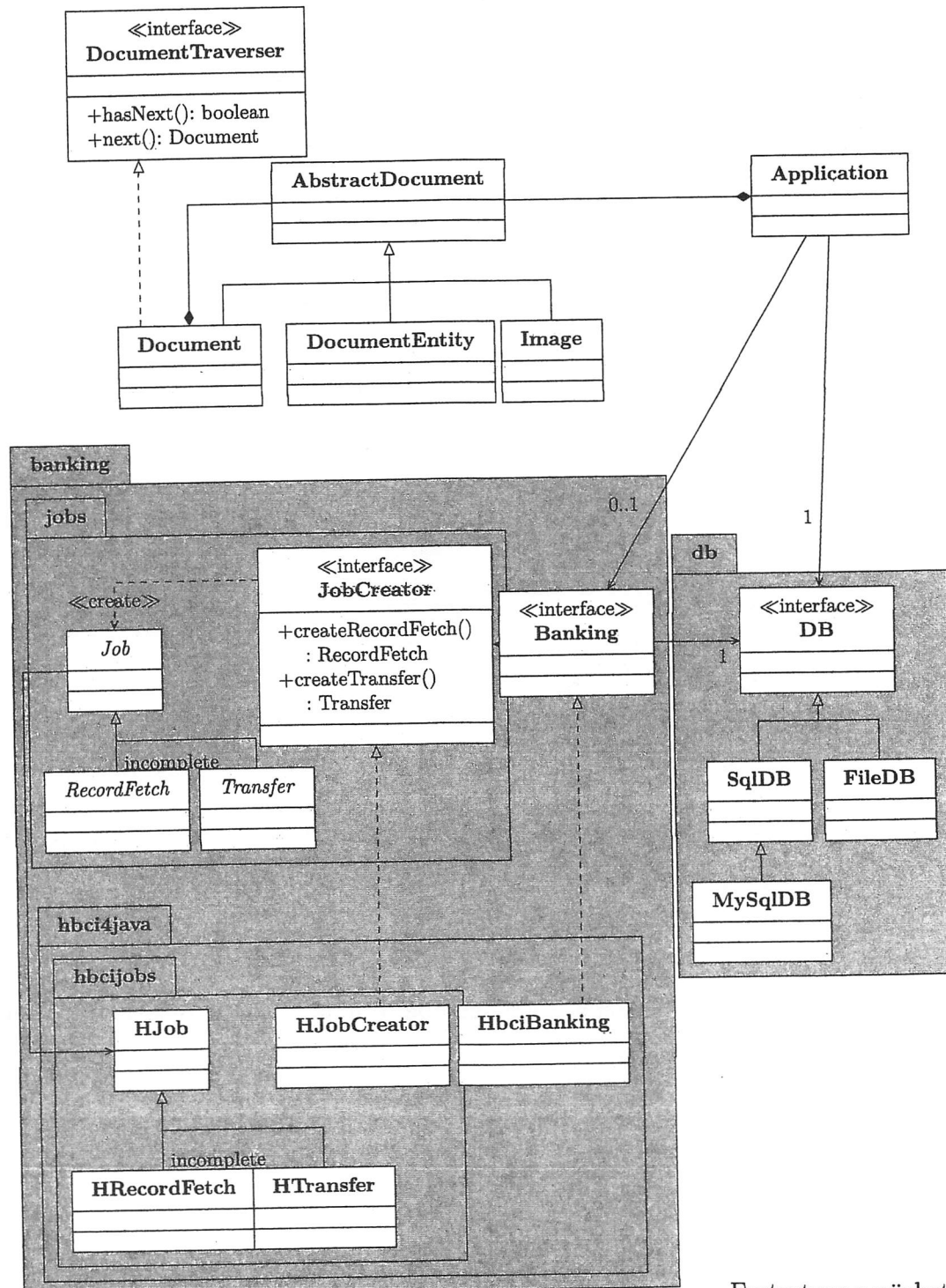
b) Verbesserungen [20 Punkte]

Benennen Sie *fünf* Möglichkeiten, wie der obige Code hinsichtlich Stil und Lesbarkeit verbessert werden kann. Geben Sie jeweils an:

- (i) warum dies im jetzigen Zustand suboptimal ist
- (ii) ein betroffenes Beispiel im Code
- (iii) wie dies behoben werden kann und
- (iv) wie die korrigierte Version aussieht.

Aufgabe 6 (Entwurfsmuster)

[20 PUNKTE]



Fortsetzung nächste Seite!

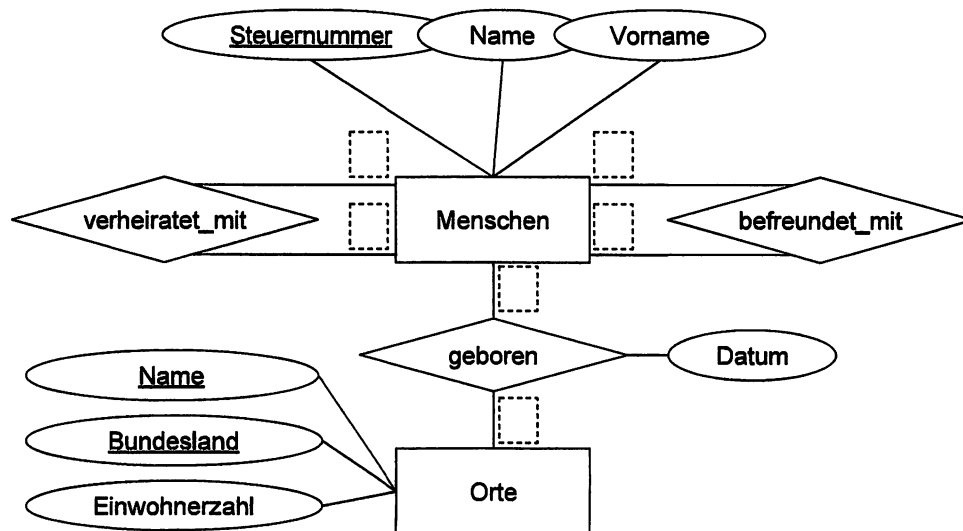
- Benennen Sie im oben zu findenden Klassendiagramm fünf verschiedene Entwurfsmuster. Geben Sie die jeweils beteiligten Klassen und deren Zuständigkeit im entsprechenden Muster an.
- Erklären Sie in maximal drei Sätzen den Zweck des Dekoratorpattern.
- Nennen Sie das Design-Prinzip, dem das Dekoratorpattern folgt.
- Nennen Sie je einen Vor- und Nachteil des Dekoratorpattern.
- Zeichnen Sie das Klassendiagramm des Dekoratorpattern. Zeichnen Sie dabei bei den dekorierten Methoden ein, wie diese aufgerufen werden.

Teilaufgabe II: Datenbanken

Aufgabe 1 (Entwurfstheorie)

[6+10+5+5 PUNKTE]

Gegeben sei folgendes ER-Diagramm:



- Übernehmen Sie das ER-Diagramm auf Ihre Bearbeitung und ergänzen Sie die Funktionalitätsangaben im Diagramm.
- Übersetzen Sie das ER-Diagramm in ein relationales Schema. – Datentypen müssen nicht angegeben werden.
- Verfeinern** Sie das Schema aus Teilaufgabe b) indem Sie die Relationen zusammenfassen.
- Geben Sie sinnvolle SQL Datentypen für Ihr verfeinertes Schema an.

Fortsetzung nächste Seite!

Aufgabe 2 (Funktionale Abhängigkeiten)

[10 PUNKTE]

Sekretäre			
PersNr	Name	Boss	Raum
4000	Freud	2125	225
4000			225
4020	Röntgen	2163	6
4020	Röntgen		26
4030	Galileo	2127	
	Freud	2137	80

Gegeben sei oben stehenden (lückenhafte) Relationenausprägung **Sekretäre** sowie die folgenden funktionalen Abhängigkeiten:

$\text{PersNr} \rightarrow \text{Name}$

$\text{PersNr}, \text{Boss} \rightarrow \text{Raum}$

Geben Sie für alle leeren Zellen Werte an, so dass keine funktionalen Abhängigkeiten verletzt werden. (Hinweis: Es gibt mehrere richtige Antworten.)

Aufgabe 3 (Relationale Anfragesprachen)

[4+4+4 PUNKTE]

Gegeben sei ein Universitätsschema (eine beispielhafte Ausprägung hängt der Klausur an - Seite 10).

- Finden Sie alle Studierenden, die **keine** Vorlesung hören. Formulieren Sie die Anfrage im **Tupelkalkül**.
- Geben Sie einen Ausdruck an, der die Relation $\neg\text{ hoeren}$ erzeugt. Diese enthält für jeden Studierenden und jede Vorlesung, die der Studierende **nicht** hört, einen Eintrag mit Matrikelnummer und Vorlesungsnummer. Formulieren Sie die Anfrage in **relationaler Algebra**.
- Finden Sie alle Studierenden, die **keine** Vorlesung hören. Formulieren Sie die Anfrage in **relationaler Algebra**.

Aufgabe 4 (Relationale Entwurfstheorie)

[2+6 PUNKTE]

Gegeben sei die Relation

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

mit den FDs

$$\begin{aligned} A &\rightarrow BCF \\ B &\rightarrow ABF \\ CD &\rightarrow EF \end{aligned}$$

Fortsetzung nächste Seite!

- a) Geben Sie alle Kandidatenschlüssel an.
- b) Überführen Sie die Relation mittels Syntheseargorithmus in die 3. NF. Geben Sie **alle Relationen** in der 3. NF an und **unterstreichen Sie in jeder einen Kandidatenschlüssel**.
– Falls Sie Zwischenschritte notieren, machen Sie das Endergebnis **klar kenntlich**.

Aufgabe 5 (Schlüssel)

[5+4+6 PUNKTE]

Gegeben sei die Relation

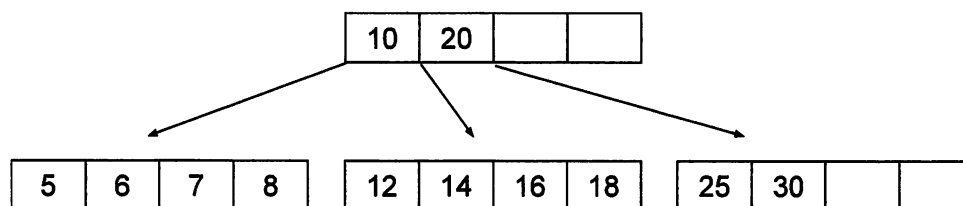
$$R : \{[A, B, C]\}$$

- a) Schreiben Sie eine SQL-Anfrage, mit der sich zeigen lässt, ob das Paar A,B ein Superschlüssel der Relation R ist. Beschreiben Sie ggf. textuell – falls nicht eindeutig ersichtlich – wie das Ergebnis Ihrer Anfrage interpretiert werden muss, um zu erkennen ob A,B ein Superschlüssel ist.
- b) Erläutern Sie den Unterschied zwischen einem Superschlüssel und einem Kandidatenschlüssel.
Tipp: Was muss gelten, damit A,B ein Kandidatenschlüssel ist und nicht nur ein Superschlüssel?
- c) Sei A,B der Kandidatenschlüssel für die Relation R. Geben Sie eine minimale Ausprägung der Relation R an, die diese Eigenschaft erfüllt.

Aufgabe 6 (Physische Datenstrukturen)

[5 PUNKTE]

Fügen Sie die Zahl 4 in den folgenden B-Baum ein.



Zeichnen Sie den vollständigen, resultierenden Baum.

Aufgabe 7 (Zehnkampf)

[4+8+6 PUNKTE]

Gegeben sei die Relation *Zehnkampf*, welche die Ergebnisse eines Zehnkampfwettkampfes verwaltet. Eine beispielhafte Ausprägung ist in nachfolgender Tabelle gegeben.

Hinweise: Jeder Athlet kann in jeder Disziplin maximal ein Ergebnis erzielen. Außerdem können Sie davon ausgehen, dass jeder Name eindeutig ist.

Fortsetzung nächste Seite!

Name	Disziplin	Leistung	Einheit	Punkte
John	100m	10.21	Sekunden	845
Peter	Hochsprung	213	Zentimeter	812
Peter	100m	10.10	Sekunden	920
Hans	100m	10.21	Sekunden	845
Hans	400m	44.12	Sekunden	910
...

- Bestimmen Sie alle funktionale Abhängigkeiten, die **sinnvollerweise** in der Relation *Zehnkampf* gelten.
- Normalisieren Sie die Relation *Zehnkampf* unter Beachtung der von Ihnen identifizierten funktionalen Abhängigkeiten. Unterstreichen Sie alle Schlüssel des resultierenden Schemas.
- Bestimmen Sie in SQL den Athleten (oder bei Punktgleichheit, die Athleten), der in der Summe am meisten Punkte in allen Disziplinen erzielt hat. Benutzen Sie dazu die noch nicht normalisierte Ausgangsrelation *Zehnkampf*.

Aufgabe 8 (SQL)

[4+4+4+5+4+5 PUNKTE]

Gegeben sei das Universitätsschema aus Aufgabe 3 (siehe Anhang - Seite 10). Formulieren Sie folgende Anfragen in SQL-92:

- Welche Vorlesungen liest der Boss des Assistenten *Platon* (**nur Vorlesungsnummer und Titel ausgeben**)?
- Welche Studierende haben sich schon in mindestens einer direkten Voraussetzung von '*Wissenschaftstheorie*' prüfen lassen?
- Wie viele Studierende hören '*Ethik*'?
- Welche Studierende sind im gleichen Semester? – Geben Sie Paare von Studierenden aus. Achten Sie darauf, dass ein/e Studierende/r mit sich selbst kein Paar bildet. – Achten Sie auch darauf, dass kein Paar doppelt ausgegeben wird: wenn das Paar *StudentA*, *StudentB* im Ergebnis enthalten ist, soll nicht auch noch das Paar *StudentB*, *StudentA* ausgegeben werden.
- In welchen Fächern ist die Durchschnittsnote schlechter als 2? Geben Sie die Vorlesungsnummer und den Titel aus.
- Finden Sie alle Paare von Studierenden (MatrNr duplikatfrei ausgeben), die mindestens **zwei** Vorlesungen gemeinsam hören.

Fortsetzung nächste Seite!

Beispielausprägung (zu Aufgabe 3 und 8)

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

Studierende		
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. 1: Beispielausprägung für eine Universitäts-Datenbank

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1

[9 PUNKTE]

Zeichnen Sie ein Gantt-Diagramm für folgenden Projektplan:

Aktivität	Abhängig von	Dauer in Wochen
A1	–	3
A2	–	6
A3	A1	4
A4	A1	5
A5	A4	4
A6	A4	4

Bestimmen Sie anschließend die Länge des Kritischen Pfades und geben Sie an, welche Arbeitspakete an ihm beteiligt sind.

Aufgabe 2

[10 PUNKTE]

- Ordnen Sie die folgenden Ergebnisse den folgenden Phasen des „klassischen“ Wasserfallmodells zu:
(A) technische Dokumentation einzelner Module; (B) fertiges System; (C) Entwurfsdokument mit Software-Bauplan; (D) Benutzerhandbuch; (E) Software-Änderungs- und -Renovierungsvorschläge; (F) Pflichtenheft
Phasen des Wasserfallmodells:
 - Anforderungsdefinition und Anforderungsaufnahme
 - System- und Softwareentwurf
 - Programmierung und Unit-Testing
 - Integration und System-Test
 - Betrieb und Wartung
- Nennen und erklären Sie kurz vier Probleme, die bei der Verwendung des Wasserfallmodells auftauchen können.

Fortsetzung nächste Seite!

Aufgabe 3

[6 PUNKTE]

Nennen und erklären Sie kurz zwei der vier objektorientierten Entwurfsprinzipien.

Aufgabe 4

[15 PUNKTE]

Erstellen Sie ein UML-Sequenzdiagramm zur Abbildung des folgenden Szenarios:

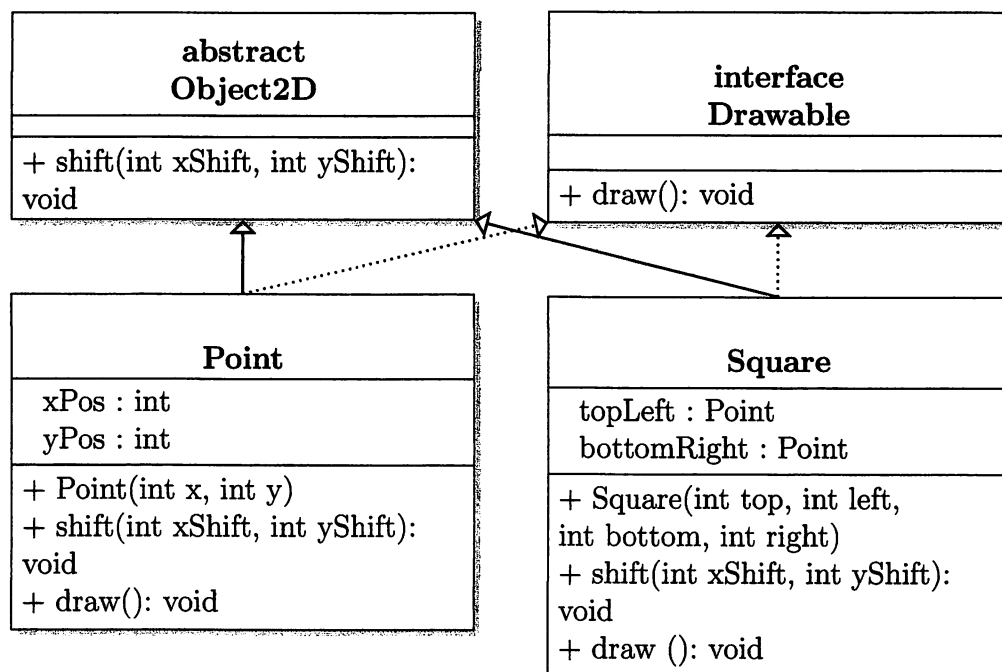
- Ein **Kunde** bestellt Karten an einem **Schalter**. Daraufhin wird er vom **Schalter** gefragt, ob er eine Ermäßigung nachweisen kann.
- Der **Kunde** sucht, leider erfolglos, seinen Studierendenausweis und sagt dem **Schalter**, dass er keinen Ermäßigungsgrund vorweisen kann.
- Der **Schalter** sagt dem **Kunden** den Preis der Karten und der **Kunde** gibt dem **Schalter** das notwendige Geld.
- Der **Schalter** erstellt die **Karten**, druckt diese aus und übergibt sie dem **Kunden**.
- Dieser geht mit den **Karten** zum **Eingang**, worauf die **Karten** zum Nachweis des Eintritts zerrissen werden.

Aufgabe 5

[36 PUNKTE]

- a) Implementieren Sie ein Programm in einer objektorientierten Programmiersprache, z. B. Java, für das folgende UML-Klassendiagramm.

Die **shift**-Methode soll die x-Position eines Objektes um **xShift** verändern und die y-Position um **yShift**. Die **draw**-Methode soll die Werte der Attribute der Klasse auf der Konsole ausgeben (– dies kann in Java mit `System.out.println ("...")` erfolgen).



Fortsetzung nächste Seite!

- b) Schreiben Sie eine Methode, die ein zweidimensionales Array aus ganzen Zahlen (Datentyp `int`) als Parameter bekommt und ein eindimensionales Array (bestehend aus ganzen Zahlen (Datentyp `int`)) zurückgibt, dessen Elemente jeweils der Summe der Einträge in der entsprechenden Zeile des zweidimensionalen Arrays entsprechen.

Achtung: Die Zeilen des zweidimensionalen Arrays können unterschiedlich lang sein.

Zur Vereinfachung sei die Signatur der Methode gegeben: `public int[] computeSum(int[] [] input)`

- c) Implementieren Sie eine einfach verkettete Liste in einer Klasse `List` (z. B. in Java), in der in jedem Listenelement ein String gespeichert wird. Die Klasse soll folgende Methoden bereitstellen:

- `void addFirst(String element)`: Diese Methode fügt ein Element am Anfang einer Liste ein.
- `void addLast(String element)`: Diese Methode hängt ein Element an das Ende der Liste an.
- `boolean exists(String element)`: Diese Methode gibt `true` zurück, wenn die Liste ein Element mit dem Inhalt `element` beinhaltet, andernfalls gibt sie `false` zurück.

Hinweis: Zwei `String`-Objekte können mittels der Funktion `equals(...)` verglichen werden.

Aufgabe 6**[9 PUNKTE]**

Bearbeiten Sie zu der gegebenen Funktion folgende Aufgaben:

1. Zeichnen Sie einen Kontrollflussgraphen für das gegebene Programm.
Hinweis: Benennen Sie die Knoten mit der zur jeweiligen Anweisung gehörigen Zeilennummer.
2. Wieviele Tests sind mindestens nötig, um eine komplette Knotenüberdeckung zu erreichen?
Geben Sie die Eingaben der notwendigen Testfälle an.

```
1 int bar(int x, int y) {  
2     int a = x;  
3     int z = a / y;  
4  
5     if(z < 5) {  
6         y = 2;  
7     } else {  
8         y = 4;  
9     }  
10    for (int i = 0; i < 10; i++) {  
11        if(a < y) {  
12            a += i;  
13        }  
14        y = x;  
15    }  
16    return a;  
17 }
```

Aufgabe 7**[8 PUNKTE]**

1. Nennen und erläutern Sie einen Vorteil und einen Nachteil bei der Verwendung der folgenden Softwaremaße, die zur Bewertung einer Software herangezogen werden können:
 - a) Zyklomatische Komplexität nach McCabe
 - b) Lines of Code
2. Was ist die Beziehung zwischen einem Code-Smell und einem Refactoring? Erklären Sie dazu beide Begriffe.
3. Kann eine zuverlässige Software trotzdem eine schlechte Qualität haben? Begründen Sie.

Fortsetzung nächste Seite!

Teilaufgabe II: Datenbanken**Aufgabe 1 (ER-Modellierung)****[16+16 PUNKTE]**

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

Gegeben seien folgende Informationen:

- Ein Wetterdienst - identifiziert durch einen eindeutigen Namen - betreibt mehrere Wetterstationen, die mit einer eindeutigen Nummer je Wetterdienst identifiziert werden können. Jede Wetterstation hat zudem mehrere Messgeräte, die wiederum pro Wetterstation einen eindeutigen Code besitzen und zudem eine Betriebsdauer.
 - Ein Wetterdienst hat eine Adresse.
 - Bei den Messgeräten wird unter anderem zwischen manuellen und digitalen Messgeräten unterschieden. Dabei kann ein Messgerät immer nur zu einer Kategorie gehören.
 - Meteorologen sind Mitarbeiter eines Wetterdienstes, haben einen Namen und werden über eine Personalnummer identifiziert. Zudem soll gespeichert werden, in welcher Wetterstation welcher Mitarbeiter zu welchem Zeitpunkt arbeitet.
 - Meteorologen können für eine Menge an Messgeräten verantwortlich sein.
 - Manuelle Messgeräte werden von Meteorologen abgelesen.
 - Ein Wettermoderator präsentiert das vorhergesagte Wetter eines Wetterdienstes für einen Fernsehsender.
 - Für einen Wettermoderator wird der eindeutige Name und die Größe gespeichert. Ein Fernsehsender wird ebenfalls über den eindeutigen Namen identifiziert.
- a) Erstellen Sie für das oben gegebene Szenario ein geeignetes ER-Diagramm. Verwenden Sie dabei – wenn angebracht – das Prinzip der Spezialisierung. Kennzeichnen Sie die Primärschlüssel der Entity-Typen, totale Teilnahmen (existenzabhängige Beziehungen) und schwache Entity-Typen. Zeichnen Sie die Funktionalitäten der Relationship-Typen in das Diagramm ein.
- b) Überführen Sie das in Teilaufgabe a) erstellte ER-Modell in ein verfeinertes relationales Schema. Kennzeichnen Sie die Schlüssel durch Unterstreichen. Datentypen müssen nicht angegeben werden. Die einzelnen Schritte müssen angegeben werden.

Fortsetzung nächste Seite!

Aufgabe 2 (SQL und relationale Algebra)**[6+4+8+4+4+12 PUNKTE]**

Gegeben sei der folgende Ausschnitt eines Schemas für die Verwaltung von Kollektionen:

```
Promi : {[
    Name : VARCHAR(255),
    Alter : INTEGER,
    Wohnort : VARCHAR(255)
]}

Kleidungsstueck : {[
    ID : INTEGER,
    Hauptbestandteil : VARCHAR(255),
    gehoert_zu : VARCHAR(255)
]}

Kollektion : {[
    Name : VARCHAR(255),
    Jahr : INTEGER,
    Saison : VARCHAR(255)
]}

hat_getragen : {[
    PromiName : VARCHAR(255),
    KleidungsstueckID : INTEGER,
    Datum : DATE
]}

promotet: {[
    PromiName : VARCHAR(255),
    KollektionName : VARCHAR(255)
]}
```

Die Tabelle *Promi* beschreibt Promis über ihren eindeutigen Namen, ihr Alter und ihren Wohnort. *Kollektion* enthält Informationen über Kollektionen, nämlich deren eindeutigen Namen, das Jahr und die Saison. Die Tabelle *promotet* verwaltet über Referenzen, welcher Promi welche Kollektion promotet. *Kleidungsstück* speichert die IDs von Kleidungsstücken zusammen mit dem Hauptbestandteil und einer Referenz auf die zugehörige Kollektion. Die Tabelle *hat_getragen* verwaltet über Referenzen, welcher Promi welches Kleidungsstück an welchem Datum getragen hat.

Beachten Sie bei der Formulierung der SQL-Anweisungen, dass die Ergebnisrelationen keine Duplikate enthalten dürfen. Sie dürfen geeignete Views definieren.

1. Schreiben Sie SQL-Anweisungen, welche die Tabelle *hat_getragen* inklusive aller benötigten Fremdschlüsselconstraints anlegt. Erläutern Sie kurz, warum die Spalte *Datum* Teil des Primärschlüssels ist.
2. Schreiben Sie eine SQL-Anweisung, welche die Namen der Promis ausgibt, die eine Sommer-Kollektion promoten (Saison ist 'Sommer').
3. Schreiben Sie eine SQL-Anweisung, die die Namen aller Promis und der Kollektionen bestimmt, welche der Promi zwar promotet, aber daraus noch kein Kleidungsstück getragen hat.
4. Bestimmen Sie für die folgenden SQL-Anweisungen die minimale und maximale Anzahl an Tupeln im Ergebnis. Beziehen Sie sich dabei auf die Größe der einzelnen Tabellen.

Fortsetzung nächste Seite!

Verwenden Sie für die Lösung folgende Notation:

— Promi — beschreibt die Größe der Tabelle *Promi*.

a)

```
SELECT k.Name
FROM Kollektion k, Kleidungsstueck kl
WHERE k.Name = kl.gehoert_zu and k.Jahr = 2018
GROUP BY k.Name
HAVING COUNT(kl.Hauptbestandteil)>10
```

b)

```
SELECT DISTINCT k.Jahr
FROM Kollektion k
WHERE k.Name IN (
    SELECT pr.KollektionName
    FROM Promi p, promotet pr
    WHERE p.Alter < 30 AND pr.PromiName = p.Name
)
```

5. Beschreiben Sie den Effekt der folgenden SQL-Anfrage in natürlicher Sprache

```
SELECT pr.KollektionName
FROM promotet pr, Promi p
WHERE pr.PromiName = p.Name
GROUP BY pr.KollektionName
HAVING COUNT(*) IN (
    SELECT MAX(anzahl)
    FROM (
        SELECT k.Name, COUNT(*) AS anzahl
        FROM Kollektion k, promotet pr
        WHERE k.Name = pr.KollektionName
        GROUP BY k.Name
    )
)
```

6. Formulieren Sie folgende SQL-Anfrage in **relationaler Algebra**. Die Lösung kann in Baum- oder in Term-Schreibweise angegeben werden, wobei eine Schreibweise genügt.

```
SELECT p.Wohnort
FROM Promi p, promotet pr, Kollektion k
WHERE p.Name = pr.PromiName
AND k.Name = pr.KollektionName
AND k.Jahr = 2018
```

- a) Konvertieren Sie zunächst die gegebene SQL-Anfrage in die zugehörige Anfrage in relationaler Algebra nach Standard-Algorithmus.
- b) Führen Sie anschließend eine relationale Optimierung durch. Beschreiben und begründen Sie dabei kurz jeden durchgeführten Schritt.

Fortsetzung nächste Seite!

Aufgabe 3 (Entwurfstheorie)**[4+10+7+6+3 PUNKTE]**

Gegeben sei folgendes relationales Schema R in erster Normalform:

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

Für R gelte folgende Menge FD funktionaler Abhängigkeiten:

$FD = \{$

$$\begin{aligned} A &\rightarrow F, \\ CEF &\rightarrow AB, \\ AE &\rightarrow B, \\ BC &\rightarrow D, \\ AF &\rightarrow C \end{aligned}$$

$\}$

1. Bestimmen Sie alle Kandidatenschlüssel/Schlüsselkandidaten von R mit FD . Begründen Sie Ihre Antwort. Begründen Sie zudem, warum es keine weiteren Kandidatenschlüssel/Schlüsselkandidaten gibt.
Hinweis: Die Angabe von Attributmengen, die keine Kandidatenschlüssel sind, führt zu Abzügen.
2. Prüfen Sie, ob R mit FD in 2NF bzw. 3NF ist.
3. Bestimmen Sie mit folgenden Schritten eine kanonische Überdeckung FD_C von FD . Begründen Sie jede Ihrer Entscheidungen:
 - a) Führen Sie eine Linksreduktion von FD durch. Geben Sie die Menge funktionaler Abhängigkeiten nach der Linksreduktion an (FD_L).
 - b) Führen Sie eine Rechtsreduktion des Ergebnisses der Linksreduktion (FD_L) durch. Geben Sie die Menge funktionaler Abhängigkeiten nach der Rechtsreduktion an (FD_R).
 - c) Bestimmen Sie eine kanonische Überdeckung FD_C von FD auf Basis des Ergebnisses der Rechtsreduktion (FD_R).
4. Zerlegen Sie R mit FD_C mithilfe des *Synthesealgorithmus* in 3NF. Geben Sie zudem alle funktionalen Abhängigkeiten der erzeugten Relationenschemata an.
5. Prüfen Sie für alle Relationen der Zerlegung aus 4., ob sie jeweils in BCNF sind.

Fortsetzung nächste Seite!

Aufgabe 4 (Transaktionen)**[16+4 PUNKTE]**

1. Betrachten Sie den folgenden Schedule S :

T_1	T_2	T_3
	$r_2(z)$	
		$w_3(y)$
	$r_2(x)$	
$w_1(x)$		
	$w_2(x)$	
		$r_3(z)$
		c_3
	$w_2(z)$	
$w_1(y)$		
c_1		
	c_2	

Geben Sie den Ausgabeschedule (einschließlich der Operationen zur Sperranforderung und -freigabe) im rigorosen Zweiphasen-Sperrprotokoll für den obigen Eingabeschedule S an.

2. Beschreiben Sie den Unterschied zwischen dem herkömmlichen Zweiphasen-Sperrprotokoll (2PL) und dem rigorosen Zweiphasen-Sperrprotokoll. Warum wird in der Praxis häufiger das rigorose Zweiphasen-Sperrprotokoll verwendet?