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

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Herbst
2018**

46116

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Softwaretechnologie/Datenbanksysteme**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 19

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

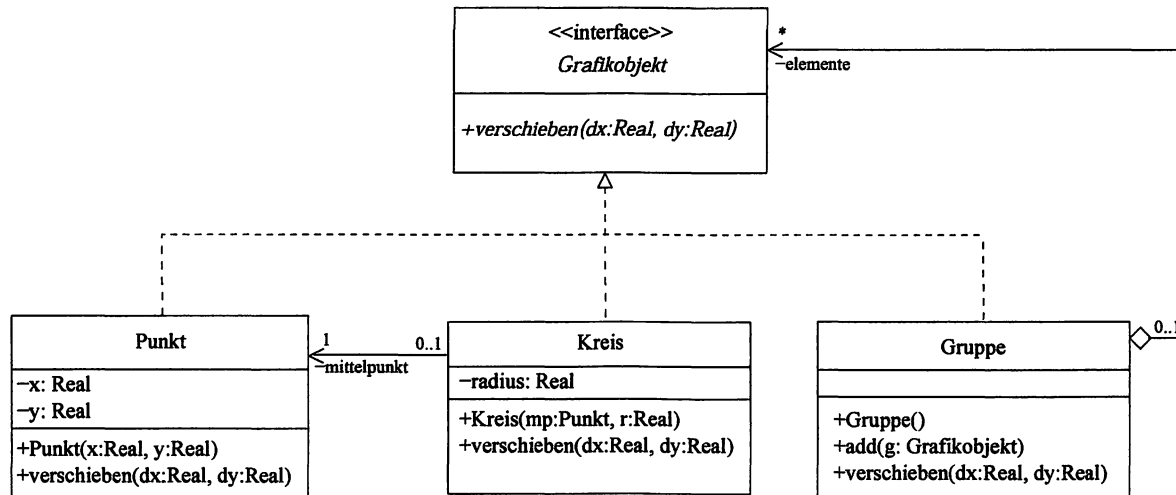
Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Softwaretechnologie

Aufgabe 1 (Entwurf von Klassendiagrammen) (19 PUNKTE)

Jeder Kaufvertrag für eine Immobilie wird von einem Notar besiegelt und zwischen einem Verkäufer und einem Käufer abgeschlossen, die beide juristische Personen mit Namen und Anschrift sind. Eine juristische Person ist entweder eine Privatperson oder ein Unternehmen mit Gewerbenummer. Immobilien sind entweder Eigentumswohnungen oder Häuser. Jede Immobilie hat eine Adresse. Eigentumswohnungen haben eine Wohnungsnummer, während Häuser eine Flurnummer haben. Notare haben einen Namen und einen Amtssitz. In jedem Kaufvertrag wird der Verkaufspreis und die Urkundennummer des Vertrags festgehalten.

Um ein System zur Verwaltung von Immobilienkäufen zu entwickeln, soll ein Datenmodell in Form eines UML-Klassendiagramms entworfen werden. Konstruieren Sie ein Klassendiagramm für den oben beschriebenen Sachverhalt mit geeigneten (ggf. abstrakten) Klassen, Vererbungsbeziehungen und Assoziationen. Assoziationen sollen geeignet benannt werden und an allen Assoziationsenden sollen passende Multiplizitäten angegeben werden.

Aufgabe 2 (Design-Pattern und objektorientierte Implementierung) (5+20+5=30 PUNKTE)

Das obige Klassendiagramm modelliert Grafikobjekte im zweidimensionalen Raum. Ein Grafikobjekt ist ein Punkt, ein Kreis oder eine Gruppe von Grafikobjekten. Jedes Grafikobjekt kann entlang eines Vektors, gegeben durch die Parameter dx und dy der Operation *verschieben*, in der Ebene verschoben werden. Die Lage eines Punktes wird durch seine x - und y -Koordinaten bestimmt, die Lage eines Kreises durch seinen Mittelpunkt und den Radius. Mit der Operation *add* kann ein Grafikobjekt zu einer Gruppe hinzugenommen werden. Eine Gruppe wird verschoben, indem alle ihre Elemente entsprechend verschoben werden. Der Datentyp *Real* steht für reelle Zahlen.

- Welches Design-Pattern wurde hier für die Modellierung von Grafikobjekten angewendet? Erläutern Sie Ihre Antwort, indem Sie den Namen des Patterns angeben und erklären, wie die für das Pattern typischen Ideen hier zum Einsatz kommen.
- Das obige Klassendiagramm ist in einer objektorientierten Programmiersprache Ihrer Wahl zu implementieren. Die verwendete Sprache ist zu nennen. Der Datentyp *Real* ist durch einen Typ für Gleitkommazahlen zu realisieren. Für die Realisierung von Mengen mit Elementen eines Typs C kann ein geeigneter Kollektionstyp `HashSet<C>` vorausgesetzt werden mit einem Konstruktor `HashSet<C>()`, der ein Objekt erzeugt, das eine leere Menge repräsentiert. Wir nehmen an, dass der Typ `HashSet<C>` eine Methode `add` zum Hinzufügen von Elementen zu einer Menge zur Verfügung stellt und dass man über alle Elemente x eines `HashSet<C>`-Objekts `hs` iterieren kann mit der Anweisung `for (C x : hs) {<Anweisungen>}`. Der Konstruktor für Punkte soll die beiden Koordinaten eines Punkts initialisieren; der Konstruktor für Kreise soll den Radius und den Mittelpunkt initialisieren. Der Konstruktor für Gruppen soll eine Gruppe erzeugen, die (zunächst) kein Grafikobjekt enthält.
- Schreiben Sie eine *main*-Methode mit Kopf `public static void main(String[] args)`, so dass am Ende der Methodenausführung folgende drei Objekte existieren: ein Punkt p mit Koordinaten $(7.0, -4.0)$, ein Kreis k mit Mittelpunkt p und Radius 3.14 und eine Gruppe, die p und k enthält.

Aufgabe 3 (Verhaltens-Modellierung mit Zustandsdiagrammen) (23 PUNKTE)

Eine Digitaluhr kann alternativ entweder die Zeit (Stunden und Minuten) oder das Datum (Tag, Monat und Jahr) anzeigen. Zu Beginn zeigt die Uhr die Zeit an. Sie besitzt drei Druckknöpfe A, B und C. Mit Knopf A kann zwischen Zeit- und Datumsanzeige hin und her gewechselt werden.

Wird die Zeit angezeigt, dann kann mit Knopf B der Reihe nach erst in einen Stundenmodus, dann in einen Minutenmodus und schließlich zurück zur Zeitanzeige gewechselt werden. Im Stundenmodus blinkt die Stundenanzeige. Mit Drücken des Knopfes C können dann die Stunden schrittweise inkrementiert werden. Im Minutenmodus blinkt die Minutenanzeige und es können mit Hilfe des Knopfes C die Minuten schrittweise inkrementiert werden.

Die Datumsfunktionen sind analog. Wird das Datum angezeigt, dann kann mit Knopf B der Reihe nach in einen Tagesmodus, Monatsmodus, Jahresmodus und schließlich zurück zur Datumsanzeige gewechselt werden. Im Tagesmodus blinkt die Tagesanzeige. Mit Drücken des Knopfes C können dann die Tage schrittweise inkrementiert werden. Analog blinken mit Eintritt in den entsprechenden Einstellmodus der Monat oder das Jahr, die dann mit Knopf C schrittweise inkrementiert werden können.

Wenn sich die Uhr in einem Einstellmodus befindet, hat das Betätigen des Knopfes A keine Wirkung. Ebenso wirkungslos ist Knopf C, wenn gerade Zeit oder Datum angezeigt wird.

Beschreiben Sie das Verhalten der Digitaluhr durch ein UML-Zustandsdiagramm. Dabei muss - gemäß der UML-Notation - unterscheidbar sein, was Ereignisse und was Aktionen sind. Deren Bedeutung soll durch die Verwendung von sprechenden Namen klar sein. Für die Inkrementierung von Stunden, Minuten, Tagen etc. brauchen keine konkreten Berechnungen angegeben werden. Der kontinuierliche Zeitfortschritt des Uhrwerks ist nicht zu modellieren.

Zustände sind, wie in der UML üblich, durch abgerundete Rechtecke darzustellen. Sie können unterteilt werden in eine obere und eine untere Hälfte, wobei der Name des Zustands in den oberen Teil und eine in dem Zustand auszuführende Aktivität in den unteren Teil einzutragen ist.

Aufgabe 4 (Verifikation funktionaler Programme) (18 PUNKTE)

Gegeben sei der folgende polymorphe Datentyp *'a list* zur Darstellung endlicher Listen mit Elementen eines beliebigen Typs *'a*.

datatype *'a list* = *nil* | *append of 'a * 'a list*

Die Länge einer Liste wird durch die folgende rekursive Funktion *length* definiert.

fun *length(nil)* = 0
| *length(append(x, L))* = *length(L)* + 1

Die Konkatenation (d.h. das Zusammenhängen) zweier Listen wird durch folgende rekursive Funktion *conc* definiert.

fun *conc(nil, L)* = L
| *conc(append(x, L1), L2)* = *append(x, conc(L1, L2))*

Es wird behauptet, dass für alle Ausdrücke L1 und L2 des Typs *'a list* folgendes gilt:

$$length(conc(L1, L2)) = length(L1) + length(L2)$$

Beweisen Sie diese Behauptung durch strukturelle Induktion gemäß des Aufbaus des Ausdrucks L1. Geben Sie zunächst an, welche Fälle für die Gestalt von L1 unterschieden werden müssen und welcher Fall der Basisfall der Induktion ist. Für den Nicht-Basisfall ist die Induktionsvoraussetzung und die Induktionsbehauptung zu nennen. Es ist genau anzugeben, wo die Induktionsvoraussetzung im Beweis verwendet wird.

Teilaufgabe II: Datenbanken**Aufgabe 1** (Vermischte Fragen) (2+2+2+2+2+2+2+2=16 PUNKTE)

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

- (a) Kann ein Tupel mehrfach in einer Relation enthalten sein?
- (b) In welcher Reihenfolge erhält man die Tupel bei einer SQL-Anfrage zurück, wenn man keine ORDER-BY-Klausel angibt?
- (c) Worin besteht der Unterschied, wenn ein Attribut vom Typ String einen leeren String enthält und wenn es NULL enthält?
- (d) Dürfen Fremdschlüsselwerte NULL sein?
- (e) Aus wie vielen SQL-Statements besteht eine Datenbanktransaktion mindestens und aus wie vielen höchstens?
- (f) Was ist der Effekt des Aufrufs von abort innerhalb einer Transaktion?
- (g) Dürfen während einer Transaktion inkonsistente Zustände auftreten?
- (h) Was muss die Recovery-Komponente eines Datenbanksystems, das ACID garantiert, beim Wiederanlauf nach einem Systemabsturz sicherstellen?

Aufgabe 2 (ER-Modellierung) (16 PUNKTE)

Im Folgenden finden Sie die Beschreibung eines Systems zur Verwaltung von Freizeitparks. 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 (= Existenzabhängigkeit, Partizipität) von Entitytypen.

Der Freizeitpark ist in mehrere Gebiete eingeteilt. Ein Gebiet hat einen eindeutigen Namen und eine Beschreibung. In jedem Gebiet gibt es eine oder mehrere Attraktionen. Diese verfügen über eine innerhalb ihres Gebiets eindeutige Nummer. Außerdem gibt es zu jeder Attraktion einen Namen, eine Beschreibung und ein oder mehrere Fotos. Der Freizeitpark hat Mitarbeiter. Zu diesen werden jeweils eine eindeutige ID, der Vorname und der Nachname gespeichert. Weiterhin hat jeder Mitarbeiter ein Geburtsdatum, das sich aus Tag, Monat und Jahr zusammensetzt. Die Arbeit im Freizeitpark ist in Schichten organisiert. Eine Schicht kann eindeutig durch das Datum und die Startzeit identifiziert werden. Jede Schicht hat weiterhin eine Dauer. Mitarbeiter können in Schichten an Attraktionen arbeiten. Dabei wird die Aufgabe gespeichert, die der Mitarbeiter übernimmt. Pro Schicht kann der selbe Mitarbeiter nur an maximal einer Attraktion arbeiten.

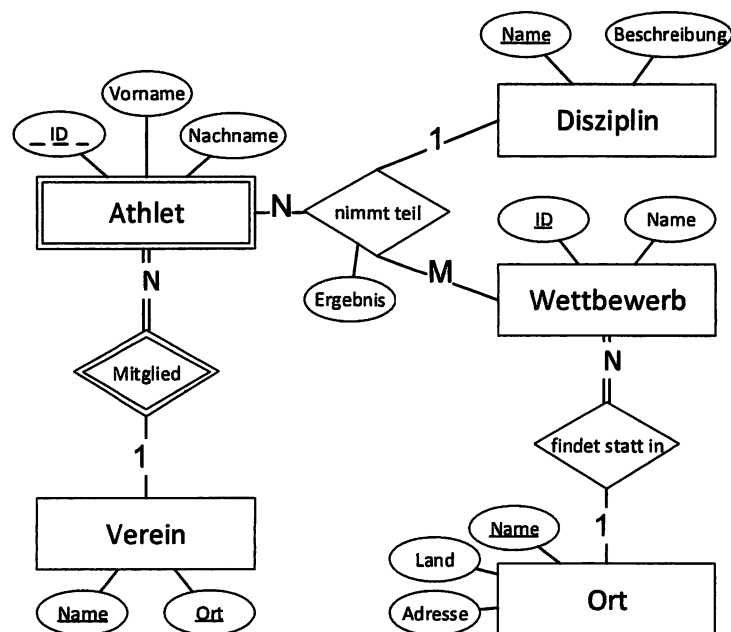
Aufgabe 3 (Relationaler Entwurf) (15 PUNKTE)

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[Relation2],  
  (Fremdschlüssel2_Attribut1, Fremdschlüssel2_Attribut2) [Relation3]);  
Attribut1 UNIQUE
```



Aufgabe 4 (SQL) (5+3+5+6+7+4=30 PUNKTE)

Gegeben sind folgende Relationen aus einem Kundenverwaltungssystem:

Kunde(ID, Vorname, Nachname, PLZ)
Produkt(GTIN, Bezeichnung, Bruttopreis, MWStSatz)
Kauf(ID[Kunde], GTIN[Produkt], Datum, Menge)

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

- (a) Schreiben Sie eine SQL-Anweisung, die die Tabelle „Kauf“ anlegt. Gehen Sie davon aus, dass die Tabellen „Kunde“ und „Produkt“ bereits existieren.
- (b) Schreiben Sie eine SQL-Anweisung, die Vorname und Nachname aller Kunden mit der Postleitzahl 20251 ausgibt, absteigend sortiert nach Nachname und bei gleichen Nachnamen, absteigend nach Vorname.
- (c) Schreiben Sie eine SQL-Anweisung, die zu jedem Einkauf mit mehr als 10 unterschiedlichen Produkten den Nachnamen des Kunden und den Bruttogesamtpreis des Einkaufs ausgibt. Ein Einkauf ist dabei definiert als Menge aller Produkte, die ein bestimmter Kunde an einem bestimmten Datum kauft.
- (d) Schreiben Sie eine SQL-Anweisung, die die GTINs aller Produkte ausgibt, die an mindestens einen in der Datenbank enthaltenen PLZ-Bereich noch nie verkauft worden sind. Als in der Datenbank enthaltener PLZ-Bereich gelten alle in der Tabelle „Kunde“ enthaltenen PLZs. Ein Produkt gilt als an einen PLZ-Bereich verkauft, sobald es von mindestens einem Kunden aus diesem PLZ-Bereich gekauft wurde. Produkte, die bisher noch gar nicht verkauft worden sind, müssen nicht berücksichtigt werden.
- (e) Schreiben Sie eine SQL-Anweisung, die die Top-Ten der am meisten verkauften Produkte ausgibt. Ausgegeben werden sollen der Rang (1 bis 10) und die Bezeichnung des Produkts. Gehen Sie davon aus, dass es keine zwei Produkte mit gleicher Verkaufszahl gibt und verwenden Sie **keine** produktspezifischen Anweisungen wie beispielsweise `rownum`, `top` oder `limit`.
- (f) Schreiben Sie eine SQL-Anweisung, die alle Produkte löscht, die noch nie gekauft wurden.

Aufgabe 5 (Normalformen) (2+2+3+6=13 PUNKTE)

- (a) Erklären Sie in ein bis zwei Sätzen, was ein Superschlüssel (Oberschlüssel) ist.
- (b) Erklären Sie in ein bis zwei Sätzen, was die funktionale Abhängigkeit $\alpha \rightarrow \beta$ auf einer Relation R für die in ihr enthaltenen Tupel bedeutet.
- (c) Erklären Sie in ein bis zwei Sätzen, wann eine Attributmenge α transitiv funktional abhängig von einer Attributmenge β ist.
- (d) Gegeben ist folgende vollständige Ausprägung (= Extension) der Relation „Prüfer“ mit dem Primärschlüssel „Prüfervummer“.

<u>Prüfervummer</u>	Nachname	Vorname	Fakultät	Dekan
9	Konrad	Franz	Medizinische Fakultät	Sonntag
10	Schmidt	Franz	Technische Fakultät	Maier
12	Schmidt	Hans	Technische Fakultät	Maier
16	Schmidt	Leon	Medizinische Fakultät	Sonntag
42	Konrad	Hans	Technische Fakultät	Maier

Geben Sie an, in welcher Normalform sich die Relation befindet. Begründen Sie weiterhin in vier bis fünf Sätzen, dass alle Bedingungen dieser Normalform erfüllt sind und dass nicht alle Bedingungen der nächsthöheren Normalform erfüllt sind.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

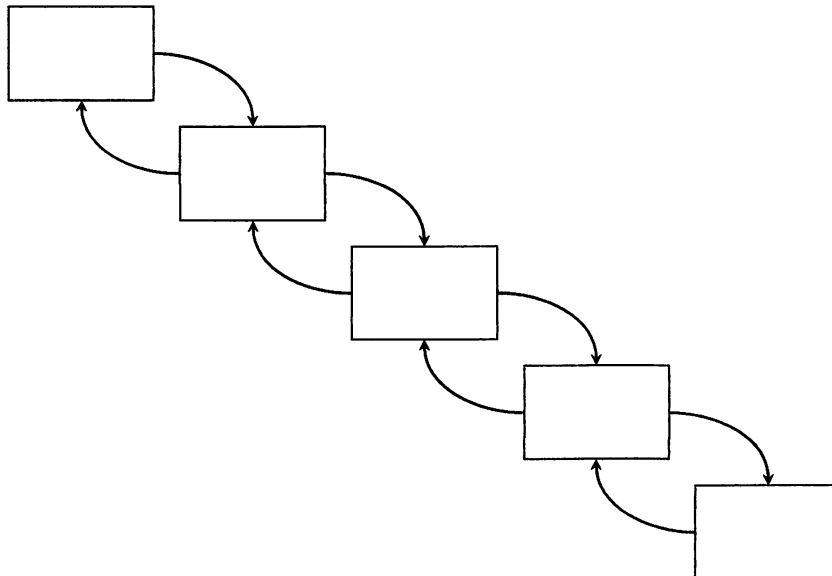
Teilaufgabe I: Softwaretechnologie

Aufgabe 1 (Projektmanagement) (13 PUNKTE)

- (a) Nennen Sie jeweils vier technische und menschliche Risiken für Projekte.
- (b) Nennen Sie die vier Stadien der Team-Entwicklung in der Reihenfolge, in der sie üblicherweise durchlebt werden.

Aufgabe 2 (Softwareprozesse) (5 PUNKTE)

Übernehmen Sie die untenstehende Grafik auf Ihre Bearbeitung und tragen Sie dort die Phasen des Wasserfallmodells ein.



Aufgabe 3 (Softwareentwurf) (7 PUNKTE)

Erläutern Sie, welche Probleme bei einem System entstehen können, welches eine starke Kopplung und eine schwache Kohäsion hat.

Definieren Sie Kohäsion und Kopplung dafür kurz!

Aufgabe 4 (Design Patterns) (9 PUNKTE)

Es soll eine Software für einen Eisladen und dessen Produkte erstellt werden:

Der Eisladen bietet drei unterschiedliche Eisarten an: einfaches Eis, Eis mit heißen Früchten und Eis mit Schokoflocken.

Zeichnen Sie ein UML-Klassendiagramm, in dem Sie die verschiedenen Eisarten als Klassen repräsentieren und außerdem das Decorator-Pattern passend verwenden.

Hinweise:

- Machen Sie Gebrauch von Vererbung.
- Methoden und Felder/Attribute müssen nicht modelliert werden.

Aufgabe 5 (Implementierung) (26 PUNKTE)

(a) Geben Sie an, welche Ausgaben das folgende Java-Programm produziert:

```
1 public class Computation {  
2     int z = 2;  
3  
4     public void computation(int x, int y) {  
5         if (x == y) {  
6             System.out.println(z * x);  
7             z = z * 3;  
8         } else if (x < y) {  
9             computation(x, (int) y / 2);  
10        }  
11        if (x > y) {  
12            System.out.println(x * y);  
13        }  
14    }  
15  
16    public static void main(String[] args) {  
17        Computation comp = new Computation();  
18        comp.computation(4, 8);  
19        comp.computation(3, 5);  
20        comp.computation(2, 2);  
21    }  
22 }  
23 }
```

- (b) Implementieren Sie einen Ringpuffer in Java, der auf einer einfach verketteten Liste basiert und zur Speicherung von ganzen Zahlen verwendet werden kann. Bei der Initialisierung des Ringpuffers soll festgelegt werden können, wie viele Elemente in dem Puffer gespeichert werden können. Soll ein Element in dem Puffer gespeichert werden, obwohl seine maximale Größe erreicht ist, ist das älteste der Elemente zu überschreiben. Generell sollen die folgenden Operationen von der Datenstruktur unterstützt werden:

- add: Fügt eine ganze Zahl in die Datenstruktur ein.
- getOldestElement: Gibt das älteste in der Liste gespeicherte Element zurück.
- numberOfElements: Gibt die Anzahl der Elemente in der Datenstruktur zurück.

Der Ringpuffer soll auf einer Liste basieren, deren Node-Klasse aus dem nachfolgendem Quelltext besteht!

```
1 public class Node {  
2  
3     int content = 0;  
4     Node next = null;  
5  
6     public Node(int content) {  
7         this.content = content;  
8     }  
9  
10    public void setNext(Node newNode) {  
11        Node next = this.next;  
12        newNode.next = next;  
13        this.next = newNode;  
14    }  
15  
16    public int getContent(){  
17        return content;  
18    }  
19  
20    public void setContent(int newValue) {  
21        this.content = newValue;  
22    }  
23  
24 }
```

Aufgabe 6 (Entwicklung und Wartung) (8 PUNKTE)

Sie haben von einem Kunden eine Fehlermeldung bekommen, dass die von Ihrem Vorgänger entwickelte Mathebibliothek einen Programmabsturz verursacht, wenn ganzzahlige Division auftritt. Sie vermuten den Fehler in der unten aufgeführten Methode. Sie müssen jetzt die Methode debuggen, um den konkreten Fehler zu identifizieren. Beschreiben Sie Ihre Schritte während des Debuggings. Nennen Sie den Fehler und beschreiben Sie passende Eingaben für das Debugging.

```
1 int dividiereGanzzahlig (int dividend, int divisor) {  
2     int result = 0;  
3     result = dividend / divisor;  
4     return result;  
5 }
```

Aufgabe 7 (Refactoring und Versionskontrolle) (4 PUNKTE)

Nennen Sie zwei Gründe, warum funktionierender Code refaktorisiert werden sollte.

Aufgabe 8 (Verifikation, Validierung und Testing) (12 PUNKTE)

(a) Gegeben ist folgende Fehlerbeschreibung von der Website eines Pizza-Lieferservices:

Wir betrachten ein Bestellformular eines Online-Pizzalieferanten. Bei der Eingabe der Adresse sieht der Kunde fehlerhafterweise nicht die Adresse, die er eingibt, sondern eine Reihe des Buchstaben "X", bspw. "XXXXXXXX"; das heißt, der Kunde tippt die Adresse blind ein. Nach der Eingabe aller weiterer notwendigen und richtigen Daten kann die Bestellung erfolgreich abgeschlossen werden und die Pizza wird entsprechend der Bestellung geliefert.

Ordnen Sie die gegebene Fehlerbeschreibung den sechs folgenden Fehlerklassen I–VI zu und begründen Sie Ihre Antwort kurz. Es können mehrere Klassen zutreffen.

(I) Transient (II) Permanent (III) Recoverable (IV) Unrecoverable (V) Non-corrupting
(VI) Corrupting

(b) Kann man durch Testen sicherstellen, dass ein Programm korrekt läuft? Begründen Sie Ihre Antwort.

Aufgabe 9 (Softwarequalität) (3 PUNKTE)

Nennen Sie drei Merkmale, an denen die Qualität einer Software festgemacht werden kann.

Aufgabe 10 (Softwaremaße) (3 PUNKTE)

Nennen Sie drei Kategorien von Softwaremaßen.

Teilaufgabe II: Datenbanken**Aufgabe 1** (Allgemeine Fragen) (3+2+2+3+1+2=13 PUNKTE)

Beantworten Sie die folgenden allgemeinen Fragen zu Datenbanksysteme.

- (a) Was sind die primären Aufgaben eines Datenbanksystems?
- (b) Was ist ein **Datenbankschema**?
- (c) Was ist ein **Datenbankzustand**?
- (d) Gegeben sei eine beliebige klassische, relationale Datenbank. Erfüllt die Datenbank eine Normalform und wenn ja, welche Normalform ist mindestens erfüllt. Begründen Sie Ihre Antwort.
- (e) Um Anfragen und Operationen effizient durchführen zu können, setzt die interne Ebene des Datenbanksystems geeignete Datenstrukturen und Speicherungsverfahren (Indexstrukturen) ein. Nennen Sie drei mögliche Anforderungen an solche Indexstrukturen.
- (f) Welches Problem kann bei redundanten Daten auftreten?
- (g) Nennen Sie vier Aggregationsfunktionen in SQL.

Aufgabe 2 (Relationale Algebra) (4+6+3=13 PUNKTE)

Gegeben seien folgende Relationen:

R:

A	B	C	D	E	F
6	8	1	7	3	7
5	3	4	4	5	7
0	6	3	0	1	7

S:

A	C	X	Z
7	8	6	1
0	3	0	0
2	3	0	5
0	6	1	6
6	7	1	7
7	1	2	2
1	8	8	0
5	1	5	5
7	3	0	2
4	8	2	7

T:

X	Y
5	3
0	5
8	6
3	6
5	7
2	8

Geben Sie die Ergebnisrelation folgender Ausdrücke der relationalen Algebra als Tabellen an. Begründen Sie Ihr Ergebnis, gegebenenfalls durch Zwischenschritte.

(a) $\sigma_{A>6}(S) \bowtie_{S.X=T.Y} \pi_Y(T)$

(b) $\pi_{A,C}(S) - (\pi_A(R) \times \pi_C(\sigma_{X=1}(S)))$

(c) $(\pi_D(R) \times \pi_E(R)) \div \pi_E(R)$

Aufgabe 3 (Anfragen) (6+4+13+15+4=42 PUNKTE)

Gegeben sei das folgende Datenbank-Schema, das für die Speicherung der Daten einer Schule entworfen wurde, zusammen mit einem Teil seiner Ausprägung. Die Primärschlüssel-Attribute sind jeweils unterstrichen.

Die Relation *Schüler* enthält allgemeine Daten zu den Schülerinnen und Schülern. Schülerinnen und Schüler nehmen an Prüfungen in verschiedenen Unterrichtsfächern teil und erhalten dadurch Noten. Diese werden in der Relation *Noten* abgespeichert. Prüfungen haben ein unterschiedliches Gewicht. Beispielsweise hat ein mündliches Ausfragen oder eine Extemporale das Gewicht 1, während eine Schulaufgabe das Gewicht 2 hat.

Schüler:

<u>SchülerID</u>	Vorname	Nachname	Klasse
1	Laura	Müller	4A
2	Linus	Schmidt	4A
3	Jonas	Schneider	4A
4	Liam	Fischer	4B
5	Tim	Weber	4B
6	Lea	Becker	4B
7	Emilia	Klein	4C
8	Julia	Wolf	4C

Noten:

<u>SchülerID</u>	<u>Schulfach</u>	Note	Gewicht	<u>Datum</u>
1	Mathematik	3	2	23.09.2017
1	Mathematik	1	1	03.10.2017
1	Mathematik	2	2	15.10.2017
1	Mathematik	4	1	11.11.2017

In dem durch die Tabellen gegebenen Datenbank-Schema sind Primärschlüssel unterstrichen und Fremdschlüssel überstrichen.

- (a) Geben Sie die SQL-Befehle an, die notwendig sind, um die oben dargestellten Tabellen in einer SQL-Datenbank anzulegen.
- (b) Entscheiden Sie jeweils, ob folgende Einfügeoperationen vom gegebenen Datenbanksystem (mit der angegebenen Ausprägung) erfolgreich verarbeitet werden können und begründen Sie Ihre Antwort kurz.
- `INSERT INTO Schüler (SchülerID, Vorname, Nachname, Klasse)
VALUES (6, "Johannes", "Schmied", "4C");`
 - `INSERT INTO Noten VALUES (9, "Chemie", 1, 2);`
- (c) Geben Sie die Befehle für die folgenden Aktionen in SQL an. Beachten Sie dabei, dass die Befehle auch noch bei Änderungen des oben gegebenen Datenbankzustandes korrekte Ergebnisse zurückliefern müssen.
- Die Schule möchte verhindern, dass in die Datenbank mehrere Kinder mit dem selben Vornamen in die gleiche Klasse kommen. Dies soll bereits auf Datenbankebene verhindert werden. Dabei sollen die Primärschlüssel nicht verändert werden. Geben Sie den Befehl an, der diese Änderung durchführt.
 - Der Schüler Tim Weber (SchülerID: 5) wechselt die Klasse. Geben Sie den SQL-Befehl an, der den genannten Schüler in die Klasse "4C" überführt.
 - Die Schülerin Laura Müller (SchülerID: 1) zieht um und wechselt die Schule. Löschen Sie die Schülerin aus der Datenbank. Nennen Sie *einen* möglichen Effekt, welcher bei der Verwendung von Primär- und Fremdschlüsseln auftreten kann.
 - Erstellen Sie eine View "DurchschnittsNoten", die die folgenden Spalten beinhaltet: Klasse, Schulfach, Durchschnittsnote
Hinweis: Beachten Sie die Gewichte der Noten.
 - Geben Sie den Befehl an, der die komplette Tabelle "Noten" löscht.
- (d) Formulieren Sie die folgenden Anfragen in SQL. Beachten Sie dabei, dass die SQL-Befehle auch noch bei Änderungen der Ausprägung die korrekten Anfrageergebnisse zurückgeben sollen.
- Gesucht ist die durchschnittliche Note, die im Fach Mathematik vergeben wird. Hinweis: Das Gewicht ist bei dieser Anfrage nicht relevant.
 - Berechnen Sie die Anzahl der Schüler, die im Fach Mathematik am 23.09.2017 eine Schulaufgabe (d.h. Gewicht=2) geschrieben haben.
 - Geben Sie die SchülerID aller Schüler zurück, die im Fach Mathematik mindestens drei mal die Schulnote 6 geschrieben haben.
 - Gesucht ist der Notendurchschnitt bezüglich jedes Faches der Klasse "4A".
- (e) Geben Sie jeweils an, welchen Ergebniswert die folgenden SQL-Befehle für die gegebene Ausprägung zurückliefern.
- `SELECT count(distinct Klasse) FROM Schüler NATURAL JOIN Noten;`
 - `SELECT count(all Klasse) FROM Noten, Schüler;`
 - `SELECT count(Note) FROM Schüler NATURAL LEFT OUTER JOIN Noten;`
 - `SELECT count(*) FROM Schüler NATURAL LEFT OUTER JOIN Noten;`

Aufgabe 4 (Tupelkalkül) (2+4=6 PUNKTE)

Gegeben sei das folgende Datenbankschema, wobei Primärschlüssel unterstrichen und Fremdschlüssel überstrichen sind. Die von einem Fremdschlüssel referenzierte Relation ist in eckigen Klammern nach dem Fremdschlüsselattribut angegeben.

Schüler(SchülerID, SVorname, SNachname, KlassenID[Klassen], Geburtsdatum)

Lehrer(LehrerID, LVorname, LNachname)

Klassen(KlassenID, Klassenstufe, Buchstabe)

Unterrichtsfächer(FachID, Name)

Noten(NotenID, SchülerID[Schüler], FachID[Unterrichtsfächer], Note, Gewicht)

LehrerUnterrichtet(LehrerID[Lehrer], KlassenID[Klassen], FachID[Unterrichtsfächer])

Formulieren Sie die folgenden Anfragen im Tupelkalkül. Geben Sie zudem das Schema aller freier Variablen an.

- (a) Bestimmen Sie die Namen (Vor- und Nachname) der Schüler, die maximal 10 Jahre alt sind. Sie können dafür eine Funktion "age" verwenden, die aus dem Geburtsdatum das Alter berechnet.
- (b) Bestimmen Sie die Nachnamen der Lehrer, die die Klasse 4A unterrichten.

Aufgabe 5 (Entity Relationship Modell) (16 PUNKTE)

Überführen Sie das Datenbankschema aus Aufgabe 4 in ein ER-Diagramm. Verwenden Sie hierfür die bereits eingezeichneten Entity-Typen und Relationship-Typen. Weisen Sie die Relationen zu und schreiben Sie deren Namen in die dazugehörigen Felder. Fügen Sie, falls erforderlich, Attribute hinzu und beschriften Sie die Beziehungen. Markieren Sie Schlüsselattribute durch unterstreichen.

Hinweis: Übernehmen Sie die untenstehende Grafik auf Ihre Bearbeitung und vervollständigen Sie sie dort.

