Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 2019

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: **15**

Bitte wenden!

Thema Nr. 1

Teilaufgabe I: Datenbanken

Aufgabe 1: ER-Modellierung

Gegeben seien folgende Informationen:

- Ein Land, beschrieben durch eine eindeutige Vorwahl und einen Namen, hat mehrere Anbieter für Mobilfunk, die wiederum einen Namen und eine innerhalb des Landes eindeutige Vorwahl besitzen.
- Ein Handy gehört zu einem Anbieter und besitzt neben einer Farbe eine nur bei dem jeweiligen Anbieter eindeutige Nummer. Handys können andere Handys anrufen, wobei der Zeitpunkt gespeichert wird.
- Personen werden anhand einer PersID identifiziert. Zudem werden Informationen über Name und Geburtsdatum gespeichert. Jeder Kontakt, identifiziert über eine ID und ein optionales Bild, entspricht genau einer Person. Zudem wird jeder Kontakt in mindestens einem Handy gespeichert.
- Eine Person kann mehrere Handys besitzen, aber ein Handy gehört höchstens einer Person.
- Handys lassen sich unter anderem in Smartphones und Tastenhandys unterscheiden, wobei ein Handy nicht beides sein kann. Für ein Smartphone wird der Speicherplatz und für ein Tastenhandy die Größe der Tasten gespeichert.

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 und schwache Entity-Typen.
Zeichnen Sie die Funktionalitäten der Relationship-Typen in das Diagramm ein.

b) Überführen Sie Ihr in Aufgabe a) erstelltes 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)

Gegeben sei der folgende Ausschnitt des Schemas für die Verwaltung der Einnahme von Medikamenten:

```
Person : {[
    ID : INTEGER,
    Name : VARCHAR(255),
    Wohnort : VARCHAR(255)
]}

Medikament : {[
    ID : INTEGER,
    Name : VARCHAR(255),
    Kosten : INTEGER,
    Wirkstoff : VARCHAR(255),
    produziert _von : INTEGER
]}

hat_Unverträglichkeit_gegen : {[
    Person : INTEGER,
    Medikament : INTEGER
]}

Hersteller : {[
    ID : INTEGER,
    Name : VARCHAR(255),
    Standort : VARCHAR(255),
    AnzahlMitarbeiter : INTEGER
]}

nimmt : {[
    Person : INTEGER,
    Medikament : INTEGER,
    von : DATE,
    bis : DATE
]}
```

Die Tabelle *Person* beschreibt Personen über eine eindeutige ID, deren Namen und Wohnort. Die Tabelle *Medikament* enthält Informationen über Medikamente, unter anderem deren Namen, Kosten, Wirkstoffe und einen Verweis auf den Hersteller dieses Medikaments. Die Tabelle *Hersteller* verwaltet verschiedene Hersteller von Medikamenten. Die Tabelle *hat_Unverträglichkeit_gegen* speichert die IDs von Personen zusammen mit den IDs von Medikamenten, gegen die diese Person eine Unverträglichkeit hat. Die Tabelle *nimmt* hingegen verwaltet die Einnahme der verschiedenen Medikamente und speichert zudem in welchem Zeitraum eine Person welches Medikament genommen hat bzw. nimmt.

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

a) Schreiben Sie SQL-Anweisungen, die für die bereits existierende Tabelle `nimmt` alle benötigten Fremdschlüsselconstraints anlegt. Erläutern Sie kurz, warum die Spalten `von` und `bis` Teil des Primärschlüssels sind.

b) Schreiben Sie eine SQL-Anweisung, welche sowohl den Namen als auch die ID von Personen und Medikamenten ausgibt, bei denen die Person das jeweilige Medikament nimmt.

c) Schreiben Sie eine SQL-Anweisung, welche die ID und den Namen der Medikamente mit den niedrigsten Kosten je Hersteller bestimmt.

d) Schreiben Sie eine SQL-Anweisung, die die Anzahl aller Personen ermittelt, die ein Medikament genommen haben, gegen welches sie eine Unverträglichkeit entwickelt haben.

Fortsetzung nächste Seite!

e) Schreiben Sie eine SQL-Anweisung, die die ID und den Namen derjenigen Personen ermittelt, die weder ein Medikament mit dem Wirkstoff `Paracetamol` noch ein Medikament mit dem Wirkstoff `Ibuprofen` genommen haben.

f) Schreiben Sie eine SQL-Anweisung, welche die Herstellernamen und die Anzahl der bekannten Unverträglichkeiten gegen Medikamente dieses Herstellers ermittelt. Das Ergebnis soll aufsteigend nach der Anzahl der bekannten Unverträglichkeiten sortiert werden.

g) Formulieren Sie eine Anfrage in **relationaler Algebra**, welche die Wohnorte aller Personen bestimmt, welche ein Medikament mit demWirkstoff `Paracetamol` nehmen oder genommen haben. Die Lösung kann als Baum- oder als Term-Schreibweise angegeben werden.

h) Formulieren Sie eine Anfrage in **relationaler Algebra**, welche die Namen aller Personen bestimmt, die von allen bekannten Herstellern, deren Standort `München` ist, Medikamente nehmen oder genommen haben. Die Lösung kann als Baum- oder als Term-Schreibweise angegeben werden.

Aufgabe 3: (Entwurfstheorie)

Gegeben sei folgendes relationales Schema R in erster Normalform:

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

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

$$ADF \rightarrow E,$$
$$BC \rightarrow AE,$$
$$D \rightarrow B,$$
$$DE \rightarrow CB,$$
$$A \rightarrow F$$

$\}$

a) Bestimmen Sie alle Kandidatenschlüssel/Schlüsselkandidaten von R mit FD.
Hinweis: Die Angabe von Attributmengen, die keine Kandidatenschlüssel sind, führt zu Abzügen.

b) Prüfen Sie, ob R mit FD in 2NF bzw. 3NF ist.

c) Bestimmen Sie mit folgenden Schritten eine kanonische Überdeckung FD_C von FD:

i. Führen Sie eine Linksreduktion von FD durch. Geben Sie die Menge funktionaler Abhängigkeiten nach der Linksreduktion an (FD_L).

ii. Führen Sie eine Rechtsreduktion des Ergebnisses der Linksreduktion (FD_L) durch. Geben Sie die Menge funktionaler Abhängigkeiten nach der Rechtsreduktion an (FD_R).

iii. Bestimmen Sie eine kanonische Überdeckung FD_C von FD auf Basis des Ergebnisses der Rechtsreduktion (FD_R).

Fortsetzung nächste Seite!

d) Zerlegen Sie R mit FD_C mithilfe des *Synthesealgorithmus* in 3NF. Geben Sie zudem alle funktionalen Abhängigkeiten der erzeugten Relationenschemata an.

e) Prüfen Sie *für alle Relationen* der Zerlegung aus d), ob sie jeweils in BCNF sind.

Aufgabe 3: (Transaktionen)

a) Nennen Sie die Phasen, die während des Ablaufs eines Zweiphasen-Sperrprotokolls auftreten und beschreiben Sie diese kurz.

b) Betrachten Sie den folgenden Schedule S:

T_1	T_2	T_3
	$r_2(x)$	
$w_1(y)$		
		$r_3(z)$
$r_1(x)$		
	$w_2(y)$	
	$r_2(z)$	
		$r_3(x)$
$w_1(z)$		
c_1		
		$w_3(x)$
		c_3
	$w_2(x)$	
	c_2	

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

Fortsetzung nächste Seite!

Teilaufgabe II: Softwaretechnik

Aufgabe 1: (Entwurfsmuster)

Gegeben sei folgender Sachverhalt: Eine Grafik ist entweder ein Kreis, ein Quadrat oder ein Dreieck. Eine Grafik kann zudem auch eine Kombination aus diesen Elementen sein. Des Weiteren können Sie aus mehreren Grafiken auch neue Grafiken zusammenbauen. Sie denken sich: Ich möchte eine Menge von Grafiken genauso wie eine einzelne Grafik behandeln können.

a) Welches Entwurfsmuster sollten Sie zur Modellierung verwenden?

b) Zeichnen Sie das entsprechende Klassendiagramm. Es reicht, nur die Klassennamen mit ihren Assoziationen und Vererbungsbeziehungen anzugeben; d.h. ohne Methoden und Attribute.

Hintergrundinformationen für die Aufgaben 2-5: Modellierung und Implementierung eines Programms

Ein autonomer Roboter in einer Montagehalle bekommt einen Auftrag, eine Menge von Objekten aus dem Lager (Material, z. B. Schrauben oder Werkzeug, z. B. Bohrer) zu holen und anschließend an seinen Ausgangspunkt zu bringen. Der Roboter hat einen Namen, der ihn eindeutig identifiziert, kennt seine aktuelle Position (x- und y-Koordinate) und hat als weitere Eigenschaft den Auftrag, der aus einer Liste von Auftragspositionen besteht, die er holen soll. Der Roboter besitzt folgende Fähigkeiten (Methoden):

- Er kann sich zu einer angegebenen Position bewegen.
- Er hat eine Methode, um einen Auftrag abzuarbeiten, d.h. alle Auftragspositionen zu holen.

Alle Objekte im Lager haben jeweils einen Namen, einen Standort mit Angabe einer Position (s. oben) und speichern außerdem die jeweils noch vorhandene Stückzahl. Es gibt zwei Typen von Objekten: Werkzeuge mit einer Methode, mit der ein einzelnes Werkzeug ausgeliehen werden kann, und Materialien mit einer Methode, mit der sie verbraucht werden, wobei eine gewünschte Stückzahl angegeben wird. Diese vorgegebenen Methoden aktualisieren die Stückzahlen der Werkzeuge (Reduktion um 1) bzw. Materialien (Reduktion um verbrauchte Stückzahl), wobei hier vereinfachend angenommen wird, dass die Stückzahlen der Werkzeuge und Materialien immer ausreichend groß sind, um die geforderten Mengen bedienen zu können (d.h. Sie können diese beiden Methoden nutzen, ohne deren Implementierung angeben zu müssen).

Ein Auftrag (z. B. "Hole Bohrer Typ B1, 100 x Schrauben M6, 10 x Schrauben M10 und 2x Blech B72") besteht aus einer Menge von Auftragspositionen. Eine Auftragsposition besteht aus dem Typ des zu holenden Objekts (Werkzeug oder Material, soll als Enumeration modelliert werden), einem Verweis auf das zu holende Objekt und der zu holenden Stückzahl (Quantität; bei Werkzeugen wird diese ignoriert, da sie immer 1 ist). Der Roboter soll über die Auftragsposition außerdem die Position bestimmen, zu der er fahren muss, um das Objekt zu holen.
Der Roboter arbeitet die Auftragspositionen in der Reihenfolge ab, indem er sich von seinem aktuellen Standpunkt immer zur am nächsten liegenden Auftragsposition bewegt, um dort das nächste Objekt zu holen. Wir gehen der Einfachheit halber davon aus, dass die Montagehalle gut aufgeräumt ist und der

Fortsetzung nächste Seite!

Roboter sich quasi entlang der Luftlinie bewegen kann, d.h. die Entfernung zwischen zwei Positionen entspricht der euklidischen Distanz (Wurzel aus der Summe der Quadrate der Differenzen der x- und y-Koordinaten der Positionen). Der Roboter soll zur Kontrolle als weitere Eigenschaft "Ergebnis" die Liste der eingesammelten Objekte zu einem Auftrag in der Reihenfolge speichern, in der er sie geholt hat, z. B. (M6 M10 B72 B1).

Aufgabe 2 (UML-Modellierung: Klassendiagramm)

Geben Sie ein UML-Klassendiagramm zu der Aufgabenstellung an.
Hinweis: Bei den Aufgaben 4 und 5 wird Konsistenz des Aktivitätsdiagramms bzw. des Codes mit dem Klassendiagramm verlangt.

Aufgabe 3 (Testgetriebene Entwicklung)

a) Erläutern Sie kurz, was man unter der Methode der testgetriebenen Entwicklung versteht.

b) Geben Sie für obige Aufgabenstellung (Abarbeitung der Aufträge durch den Roboter) einen Testfall für eine typische Situation an (d. h. das Einsammeln von 4 Objekten an unterschiedlichen Positionen). Spezifizieren Sie als Input alle für die Abarbeitung eines Auftrages relevanten Eingabe- und Klassen-Parameter sowie den vollständigen und korrekten Output.

Aufgabe 4 (Pseudocode und Aktivitätsdiagramm)

a) Geben Sie für alle Methoden, die zur Abarbeitung des Auftrages für den Roboter erforderlich sind, Pseudocode an. Nutzen Sie (im Klassendiagramm definierte) Hilfsmethoden, um den Code übersichtlicher zu gestalten. (Verwenden Sie statt einer komplizierten Methode mehrere einfachere Methoden, die sich aufrufen.)

   Achten Sie auf die Konsistenz zum Klassendiagramm (Inkonsistenzen führen zu Punktabzügen).

b) Überführen Sie den Pseudocode der Hauptmethode zur Abarbeitung eines Auftrags in ein syntaktisch korrektes UML-Aktivitätsdiagramm. Schleifen und Verzweigungen müssen durch strukturierte Knoten dargestellt werden (d. h. kein graphisches "goto").

Aufgabe 5 (Objektorientierte Programmierung)

Geben Sie korrekten Code für die Abarbeitung eines Auftrages durch den Roboter in einer objektorientierten Programmiersprache Ihrer Wahl an. Sie sollen nur den Code für die Methoden angeben. Sie brauchen keinen Code für Klassendefinitionen angeben, sondern können sich auf das UML-Klassendiagramm aus Aufgabe 2 beziehen.

Thema Nr. 2

Teilaufgabe I: Datenbanken

Aufgabe 1: (Allgemeine Fragen)

a) Das ACID-Prinzip fordert unter anderem die Atomarität und die Isolation einer Transaktion. Beschreiben Sie kurz die Bedeutung dieser Begriffe.

b) Was versteht man unter physischer Datenunabhängigkeit?

c) In welche drei Ebenen unterteilt die 3-Ebenen-Architektur Datenbanksysteme?

d) Definieren Sie, was ein Schlüssel ist.

e) Gegeben ist eine Relation $R(A, B, C)$, deren einziger Schlüsselkandidat A ist. Nennen Sie zwei Superschlüssel.

f) Gegeben seien 2 Relationen $R(A, B, C)$ und $S(C, D, E)$. Die Relation R enthalte 9 Tupel und die Relation S enthalte 7 Tupel. Sei p ein beliebiges Selektionsprädikat. Gegeben seien außerdem folgende Anfragen:

Q_1:

```
SELECT *
FROM R NATURAL JOIN S
WHERE p;
```

Q_2:

```
SELECT DISTINCT A
FROM R LEFT OUTER JOIN S ON R.C=S.C;
```

- Wieviele Ergebnistupel liefert die Anfrage Q_1 **mindestens**?
- Wieviele Ergebnistupel liefert die Anfrage Q_1 **höchstens**?
- Wieviele Ergebnistupel liefert die Anfrage Q_2 **mindestens**?
- Wieviele Ergebnistupel liefert die Anfrage Q_2 **höchstens**?

g) Erläutern Sie, was es bedeutet, wenn ein Attribut prim ist.

h) Nennen Sie die drei Armstrong-Axiome, die dazu dienen aus einer Menge von funktionalen Abhängigkeiten, die auf einer Relation gelten, weitere funktionale Abhängigkeiten abzuleiten.

i) Nennen Sie die höchste Normalform, der die Relation $R(A, B, C, D)$ mit den Abhängigkeiten $A \rightarrow C$ und $A, B \rightarrow D$ entspricht. Begründen Sie Ihre Antwort.

j) Gegeben seien die Transaktionen T_1, T_2 und T_3. Wie viele mögliche verschiedene **serialisierbare** Schedules gibt es *mindestens*? (Geben Sie Ihren Rechenweg an.)

Fortsetzung nächste Seite!

Aufgabe 2: (Relationales Modell)

Das Direktorat verliert bald den Überblick über die ganzen Smartphones, die die Lehrer von ihren Schülern während des Unterrichts konfiszieren mussten. Es wurde bei jedem Smartphone aufgeschrieben, welcher Lehrer es welchem Schüler an welchem Datum abgenommen hat. (Es kann angenommen werden, dass sich jeder Schüler nur einmal pro Tag erwischen lässt.) Um die Smartphones wieder dem jeweils richtigen Schüler zurückgeben zu können, soll deshalb eine Datenbank auf der Basis der folgenden Informationen erstellt werden. Die Schüler können durch die Kombination aus ihrem Nachnamen und Vornamen eindeutig identifiziert werden. Auch der Notenschnitt jeden Schülers ist im Lehrerkollegium bekannt.

Die Lehrer können anhand ihres Nachnamens eindeutig identifiziert werden und unterrichten jeweils drei Fächer.

Da der Informatiklehrer alles über die neuesten Trends wissen möchte, werden von den Smartphones jeweils die Modellnummer, sowie der RAM und die Bildschirmdiagonale aufgezeichnet. Um die Smartphones nicht zu verwechseln, wird jedem eine schulinterne ID zugewiesen. Auch welche Apps auf welchen Smartphones installiert sind, wird vermerkt. Es wird jeweils ihr eindeutiger Name, sowie die Bewertung im Store und ihr Preis notiert.

Erstellen Sie ein vollständiges Entity-Relationship-Diagramm für obige Datenbank. Vergessen Sie nicht, Schlüssel als solche durch Unterstreichen zu markieren und alle Funktionalitäten anzugeben.

Aufgabe 3: (Relationale Algebra)

Gegeben seien folgende Relationen:

X

A	**B**	**C**
1	2	3
2	3	4
4	3	3
1	2	2
2	3	2
3	3	2
3	1	2
2	2	1
1	1	1

Y

B	**C**	**D**
2	3	1
1	1	3
2	1	3
3	2	3
2	2	3
1	3	2

Fortsetzung nächste Seite!

Geben Sie die Ergebnisrelationen folgender Ausdrücke der relationalen Algebra als Tabellen an; machen Sie Ihren Rechenweg kenntlich.

- $\sigma_{A=2}(X) \bowtie Y$
- $(\pi_{B,C}(X) - \pi_{B,C}(Y)) \bowtie X$
- $\pi_A(X) \underset{A>D}{\bowtie} \sigma_{C=1}(Y)$
- $\pi_{A,C}(X) \div \pi_C(Y)$

Aufgabe 4: (Tupelkalkül und Bereichskalkül)

Gegeben sei das folgende Datenbankschema, welches für die Problemstellung aus Aufgabe 2 entwickelt wurde. Primärschlüssel sind unterstrichen, Fremdschlüssel sind überstrichen und der Text in den darauf folgenden eckigen Klammern benennt die Relation, auf die verwiesen wird.

Lehrer (Name, Fach1, Fach2, Fach3)
Schüler (Vorname, Nachname, Notenschnitt)
Smartphone (ID, Modellnr, RAM, Bildschirmdiagonale)
App (Name, Bewertung, Preis)
Eingesammelt (Vorname [Schüler], Nachname [Schüler], Name [Lehrer], ID [Smartphone], Datum)
Installiert (ID [Smartphone], Name [App])

Formulieren Sie die folgenden Anfragen a) bis c) im Tupelkalkül. Vergessen Sie dabei nicht, das Schema anzugeben. Formulieren Sie Aufgabe d) sowohl im Tupel- als auch im Bereichskalkül. Datumsvergleiche können Sie mit >, ≥, <, ≤ oder = angeben.

a) Geben Sie die Vornamen aller Schüler an, von denen Lehrer Lämpel am 16.10.2017 ein Smartphone eingesammelt hat.

b) Geben Sie die Namen aller Apps an, die auf den Smartphones waren, die vom Schüler Max Moritz eingesammelt wurden.

c) Geben Sie die Namen und alle Fächer der Lehrer an, die Smartphones mit einer Bildschirmdiagonale von unter 5,5" eingesammelt haben.

d) Sowohl im Tupel- ALS AUCH im Bereichskalkül: Geben Sie Namen und Hauptfach (also Fach1) der Lehrer an, die Smartphones von Schülern eingesammelt haben, die einen schlechteren Notenschnitt als 3,0 haben.

Fortsetzung nächste Seite!

Aufgabe 5: (Anfragen)

Wir betrachten erneut das gegebene Schema aus Aufgabe 4. Primärschlüssel sind unterstrichen, Fremdschlüssel sind überstrichen und der Text in den darauf folgenden eckigen Klammern benennt die Relation, auf die verwiesen wird.

Der Notenschnitt, der Preis und die Bewertung werden als Kommazahl dargestellt, wobei die Bewertung die Anzahl der Sterne angibt, also maximal 5 und mindestens 0. Die Modellnummer kann sowohl aus Zahlen und Buchstaben bestehen, ist jedoch nie länger als 50 Zeichen. ID, RAM, Bildschirmdiagonale und Datum sind ganze Zahlen. Die restlichen Attribute sind Strings der Länge 15.

Lehrer (Name, Fach1, Fach2, Fach3)
Schüler (Vorname, Nachname, Notenschnitt)
Smartphone (ID, Modellnr, RAM, Bildschirmdiagonale)
App (Name, Bewertung, Preis)
Eingesammelt (Vorname [Schüler], Nachname [Schüler], Name [Lehrer], ID [Smartphone], Datum)
Installiert (ID [Smartphone], Name [App])

a) Geben Sie die Anweisung in **SQL-DDL** an, die notwendig ist, um die Relation 'App' zu erzeugen.

b) Geben Sie die Anweisung in **SQL-DDL** an, die notwendig ist, um die Relation 'Installiert' zu erzeugen.

c) Formulieren Sie die folgende Anfrage in **SQL**:

Gesucht sind die Namen der Apps zusammen mit ihrer Bewertung, die auf den Smartphones installiert sind, die Lehrer Keating eingesammelt hat. Sortieren Sie das Ergebnis nach Bewertung absteigend.

Hinweis: Achten Sie auf gleichnamige Attribute.

d) Formulieren Sie die folgende Anfrage in **SQL**:

Gesucht ist der durchschnittliche Notenschnitt der Schüler, denen ein iPhone 6s abgenommen wurde. Ein iPhone 6s kann A1633 als Modellnummer haben oder A1688.

e) Formulieren Sie die folgende Anfrage in **SQL**:
Gesucht ist die Modellnummer der Smartphones, die durchschnittlich die meisten Apps installiert haben.

Tipp: Die Verwendung von Views kann die Aufgabe vereinfachen.

Fortsetzung nächste Seite!

Aufgabe 6: (Normalisierung)

Gegeben sei das Relationsschema $R(A, B, C, D, E, F)$, sowie die Menge der zugehörigen funktionalen Abhängigkeiten FD:

- $B \rightarrow F$
- $CD \rightarrow E$
- $C \rightarrow A$
- $CD \rightarrow A$
- $D \rightarrow F$
- $D \rightarrow B$

a) Bestimmen Sie den Schlüsselkandidaten der Relation R und begründen Sie, warum es keine weiteren Schlüsselkandidaten gibt.

b) Überführen Sie das Relationsschema R mit Hilfe des Synthesealgorithmus in die dritte Normalform. Führen Sie hierfür jeden der vier Schritte durch und kennzeichnen Sie Stellen, bei denen nichts zu tun ist. Benennen Sie alle Schritte und begründen Sie eventuelle Reduktionen.

Fortsetzung nächste Seite!

Teilaufgabe II:

Aufgabe 1: (Test-getriebene Entwicklung)

Bewerten Sie die folgenden Aussagen und nehmen Sie jeweils Stellung.

a) Tests zuerst

In test-getriebener Softwareentwicklung wird der Test vor der zu testenden Funktion programmiert.

b) Komponententests

Komponententests sind immer White-Box-Tests und nie Black-Box-Tests.

c) Akzeptanztests

Akzeptanz- resp. Funktionstests sind immer Black-Box-Tests und nie White-Box-Tests.

d) Fehler

Ein fehlgeschlagener Test und ein Testausführungsfehler bezeichnen denselben Sachverhalt.

e) Test Suiten

Tests können hierarchisch strukturiert werden, so dass mit einem Befehl das gesamte zu testende System getestet werden kann.

Aufgabe 2: (Objektorientierte Modellierung)

Eine Pizzeria möchte mit dem Webdienst PizzaNow anbieten, Pizzen online bestellen zu können. Es gilt die folgende Beschreibung:

„Ein Kunde kann mehrere Pizzen in PizzaNow zu einer Bestellung zusammenstellen. Nachdem er die Bestellung abgeschickt hat, wird die Pizzeria über diese Bestellung informiert. Dort wird die Bestellung bestätigt und der Kunde erhält eine Bestätigung mit Zahlungsinformationen. Diese müssen vom jeweiligen Kreditinstitut überprüft und bestätigt werden. Danach erhält der Kunde eine Bestätigung über den erfolgreichen Bezahlvorgang und die Pizzeria beginnt nach der Bestätigung der Bezahlung die Herstellung und im Anschluss die Auslieferung der Bestellung."

Modellieren Sie den dargestellten Prozess mit Hilfe eines UML-Sequenzdiagramms.

Fortsetzung nächste Seite!

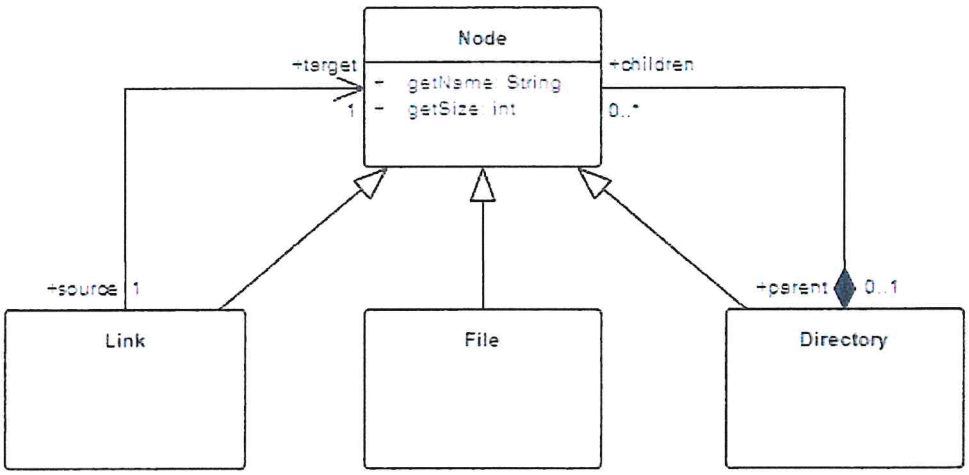
Aufgabe 3: (Objektorientierte Modellierung)

a) Nehmen Sie an, dass in einem System zur Durchführung von Umfragen die drei Anwendungsfälle „Teilnahme", „Fragen beantworten" und „Registrierung für Verlosung" gegeben seien. Alle Anwendungsfälle sind einem Teilnehmer (an einer Umfrage) zugänglich. Stellen Sie die drei Anwendungsfälle in einem UML-Anwendungsfalldiagramm („use case diagram") dar. Verwenden Sie dabei sowohl die <<extend>> - als auch die <<include>> -Beziehung und beschreiben Sie die Bedeutung der Beziehungen für die Existenz des „Teilnahme"-Anwendungsfalls.

b) Welche Informationen sollte eine typische Spezifikation eines Anwendungsfalls enthalten?

Aufgabe 4: (Objektorientierte Modellierung)

Das im Folgenden dargestellte Diagramm ist das von Aufgaben 4a)-e) referenzierte Diagramm. Das Diagramm stellt ein Klassenmodell eines Dateisystems dar.



a) Entwurfsmuster

Benennen Sie alle Anwendungen von Entwurfsmustern („design patterns"), die explizit im Diagramm erkennbar sind. (Falschmeldungen zählen negativ.)

b) Verweise

Ein Link-Objekt verweist mittels target-Referenz auf weitere Objekte. Benennen Sie die Klassen aller möglichen Objekte, die gemäß Diagramm referenziert werden können.

Fortsetzung nächste Seite!

c) Verzeichnisbaum

Benennen Sie die minimale Tiefe sowie die maximale Tiefe des durch Directory-Objekte aufgespannten Verzeichnisbaums.

d) Erweiterung

Wie würden Sie mit diesem Entwurf vereinbar eine Klasse Gerätetreiber einführen, welche externe Dateisysteme in den Verzeichnisbaum einbindet? Begründen Sie Ihre Entscheidung.

e) UML-Collaboration

Nehmen Sie an, dass Sie diesen Entwurf um eine UML-Collaboration zur Modellierung von Eltern-Kind-Beziehungen erweitern. Benennen Sie die Klasse, an welche Sie die Kind-Rolle binden würden, und begründen Sie Ihre Entscheidung.