

Bereich	Berufsnummer			IHK-Nummer	Prüflingsnummer		
5 5	1	1	9	6			

Sp. 1-2 Sp. 3-6 Sp. 7-9 Sp. 10-14



Termin: Mittwoch, 4. Mai 2022

Abschlussprüfung Sommer 2022

1196

1

Ganzheitliche Aufgabe I
Fachqualifikationen

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung (AO 1997)

5 Handlungsschritte
mit Belegsatz
90 Minuten Prüfungszeit
100 Punkte

Hinweis:

Bei der Bearbeitung der Aufgaben ist von einem gewöhnlichen Geschäftsbetrieb auszugehen, der **nicht** durch die COVID-19-Pandemie beeinflusst bzw. durch entsprechende behördliche Verfügungen eingeschränkt ist.

Bearbeitungshinweise

- Der vorliegende Aufgabensatz besteht aus insgesamt 5 Handlungsschritten zu je 25 Punkten.
In der Prüfung zu bearbeiten sind 4 Handlungsschritte, die vom Prüfungsteilnehmer frei gewählt werden können.
Der nicht bearbeitete Handlungsschritt ist durch Streichung des Aufgabentextes im Aufgabensatz und unten mit dem Vermerk „Nicht bearbeiteter Handlungsschritt: Nr. ...“ an Stelle einer Lösungsniederschrift deutlich zu kennzeichnen. Erfolgt eine solche Kennzeichnung nicht oder nicht eindeutig, gilt der 5. Handlungsschritt als nicht bearbeitet.
- Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-Nr. in die oben stehenden Felder ein.
- Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
- Halten Sie sich bei der Bearbeitung der Aufgaben genau an die **Vorgaben der Aufgabenstellung** zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier Angaben bewertet.
- Tragen Sie die frei zu formulierenden **Antworten dieser offenen Aufgabenstellungen** in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
- Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
- Verwenden Sie nur einen Kugelschreiber und schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
- Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
- Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
- Für **Nebenrechnungen/Hilfsaufzeichnungen** können Sie das im Aufgabensatz enthaltene Konzeptpapier verwenden. Dieses muss vor Bearbeitung der Aufgaben herausgetrennt werden. Bewertet werden jedoch nur Ihre Eintragungen im Aufgabensatz.

Wird vom Korrektor ausgefüllt!

Bewertung

Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen. Für den abgewählten Handlungsschritt ist anstatt der Punktzahl die Buchstabenkombination „AA“ in die Kästchen einzutragen.

Spalte 1 - 14 s. o.	1. Handlungsschritt 15 16	2. Handlungsschritt 17 18	3. Handlungsschritt 19 20	4. Handlungsschritt 21 22	5. Handlungsschritt 23 24	
Gesamtpunktzahl	26	27	28	Prüfungszeit		

Prüfungszeit
25
Die entsprechende Ziffer (1, 2 oder 3) finden Sie in der Abfrage nach der Prüfungszeit im Anschluss an die letzte Aufgabe.

Unterschrift

Die Handlungsschritte 1 bis 5 beziehen sich auf die folgende Ausgangssituation:

Die Praxisgemeinschaft ProPatient plant für Pandemiefälle ein Terminverwaltungssystem für Impfungen und Tests. Die IT-Antos GmbH übernimmt die Realisierung des Projekts.

Sie sollen vier der folgenden fünf Aufgaben in diesem Projekt erledigen:

- Ein Zustandsdiagramm für Terminvergabe erstellen
- Eine Statistische Auswertung eines zweidimensionalen Terminarrays implementieren
- Strategiepattern für Abrechnung Kassen- / Privatpatient bearbeiten
- Ein ER - Modell zur Speicherung der Daten für den Personaleinsatz anlegen
- SQL-Anweisungen für eine Datenbank formulieren

1. Handlungsschritt (25 Punkte)

Zur Terminverwaltung des Impfzentrums kann ein Termin für ein Zeitfenster gebucht werden. Jedes Zeitfenster kann einen der folgenden Zustände annehmen:

Zustand	
Grün	Ein Zeitfenster bekommt nach Einrichtung diesen Zustand, falls in dem Zeitraum ein Termin grundsätzlich auswählbar ist. Ein Zeitfenster ist im Zustand „grün“, solange noch wenigstens 10 Buchungen möglich sind.
Rot	Ein Zeitfenster ist im Zustand „rot“, sofern 1 bis 9 Buchungen möglich sind.
gesperrt	Diesen Zustand erhält ein Zeitfenster initial, wenn für diesen Zeitraum generell kein Termin buchbar ist, oder wenn im Zustand „rot“ der letzte mögliche Termin gebucht wird.

Eine Buchung/Stornierung kann immer nur für einen einzigen Termin und eine Person erfolgen.

Erstellen Sie (auf Seite 3) für die beschriebene Situation ein Zustandsdiagramm für ein Zeitfenster.

2. Handlungsschritt (25 Punkte)

Korrekturrand

Die Praxisgemeinschaft ProPatient benötigt statistische Auswertungen für die vergebenen Termine.

Terminliste 2022 (Auszug)

datum	zeit	storniert	stattgefunden
...
25.02.2022	16:30	ja	nein
25.02.2022	16:30	nein	nein
25.02.2022	16:30	nein	ja
25.02.2022	16:30	nein	ja
25.02.2022	16:30	ja	nein
25.02.2022	16:30	nein	ja
26.02.2022	09:00	nein	ja
...

Ergebnisliste (Auszug)

KW	Storniert	Stattgefunden	Ausfall
1	0	0	0
2	12	208	4
3	0	0	0
4	3	190	7
5	0	212	3
6	2	168	0
...
52	0	0	0
Jahr	87	867	53

Es liegt pro Jahr eine nach Datum und Zeit sortierte Liste vom Typ DATENSATZ mit gebuchten Terminen vor (siehe Auszug), für die eine Auswertung erstellt und ausgegeben werden soll:

- Pro Kalenderwoche soll die Anzahl der stornierten, der stattgefundenen und der ausgefallenen Termine aufsummiert werden
- Kalenderwochen ohne gebuchte Termine (alle Summen 0, z. B. KW 1, KW 3) sollen ebenfalls ausgegeben werden
- Am Ende sollen die entsprechenden Jahressummen der stornierten, stattgefunden und ausgefallenen Termine ausgegeben werden
- Hinweise:
 - *storniert* und *stattgefunden* können nicht beide den Wert **ja** haben
 - als *ausgefallen* gilt ein Termin, wenn *storniert* und *stattgefunden* beide den Wert **nein** haben
 - Folgende Datensatzstruktur steht zur Verfügung:

STRUKTUR DATENSATZ

```
datum: DATE,
zeit: TIME,
storniert: BOOL,
stattgefunden: BOOL
```

ENDE STRUKTUR

- Folgende Funktionen stehen zur Verfügung:

Funktion	Beschreibung
PROZEDUR printKopf()	Ausgabe des Listenkopfes
PROZEDUR printKW(kw, storniert, stattgefunden, ausgefallen: INTEGER)	Ausgabe Kalenderwochenzeile
PROZEDUR printJahr(storniert, stattgefunden, ausgefallen: INTEGER)	Ausgabe Jahreszeile
FUNCTION GetKW(dt: DATUM): INTEGER	Rückgabe der Kalenderwochennummer zu einem angegebenen Datum
FUNCTION GetAnzahlKW(jahr: STRING): INTEGER	Liefert die Anzahl der Kalenderwochen im angegebenen Jahr (52 oder 53)

Entwerfen Sie eine Funktion printList als Struktogramm oder Programmablaufplan oder Pseudocode die die oben aufgeführten Punkte erfüllt.

PROZEDUR printList(datenliste: LISTE<DATENSATZ>, jahr: INTEGER)

3. Handlungsschritt (25 Punkte)

Korrekturrand

Die Praxisgemeinschaft ProPatient fakturiert auf ihren Rechnungen für Privatpatienten verschiedene Leistungsarten. Der Betrag einer Rechnungsposition hängt jedoch nicht nur vom Leistungspreis, sondern auch vom Aufwand, der zur Erbringung der Leistung nötig war, ab. Zur Verwaltung soll ein objektorientiertes Programm entwickelt werden.

Beispielrechnung

Ziffer GOÄ	Beschreibung	Preis	Betrag
1	Beratung, auch telefonisch	4,66 EUR	10,72 EUR
5	Symptombezogene Untersuchung	4,66 EUR	10,72 EUR
298	Entnahme und ggf. Aufbereitung von Abstrichmaterial zur mikrobiologischen Untersuchung – ggf. einschließlich Fixierung mit erhöhtem Aufwand	2,33 EUR	8,15 EUR
A 4668	Identifizierung von Virus-Antigenen durch Immunoblotting, je Untersuchung, SARS-CoV-2 Rapid Antigen Test	19,24 EUR	22,12 EUR
A 245	Erfüllung aufwändiger Hygienemaßnahmen im Rahmen der Covid-19-Pandemie, je Sitzung	6,14 EUR	6,14 EUR
Summe			57,85 EUR

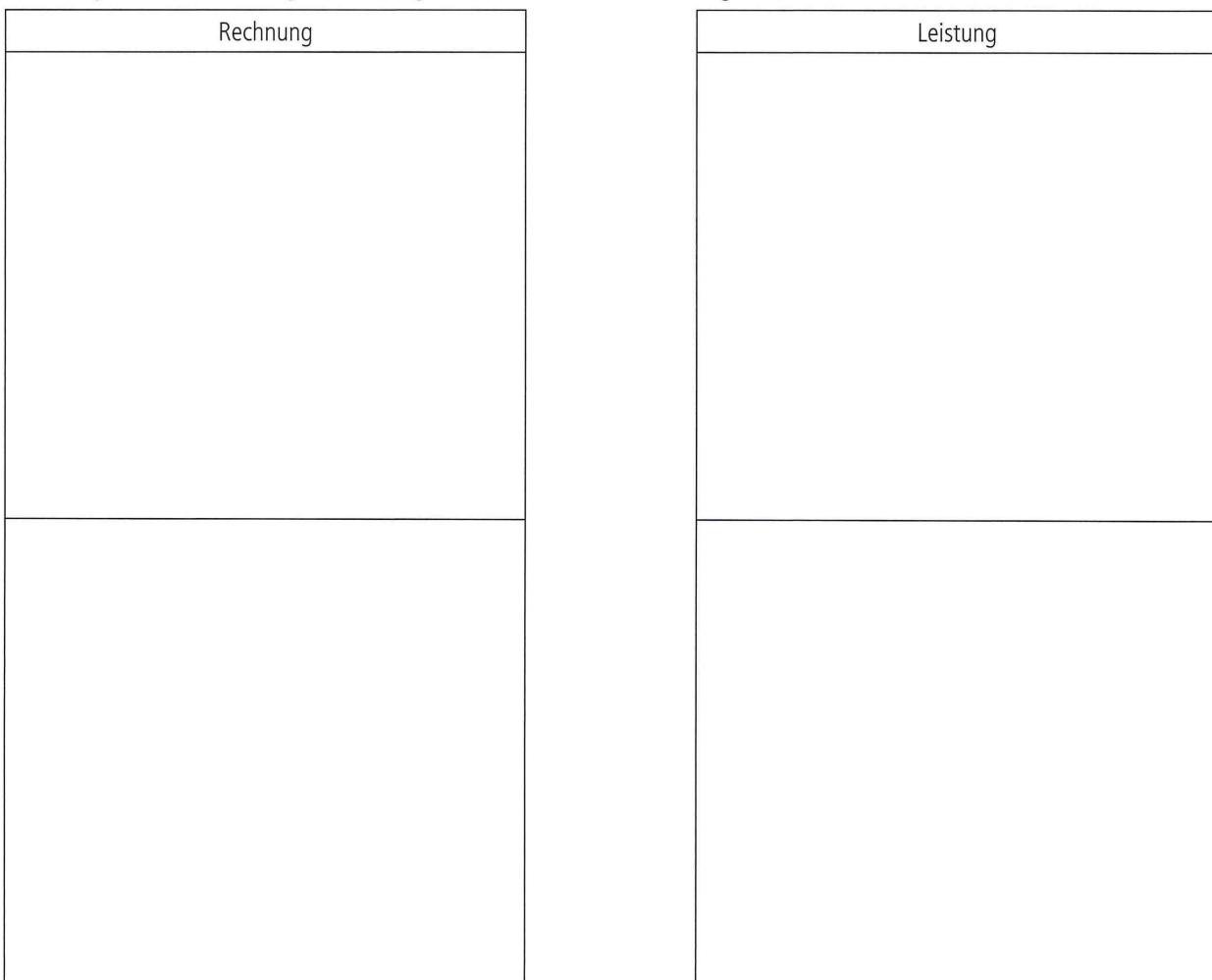
a) Folgende Anforderungen sind gegeben:

- Eine *Leistung* soll eine Beschreibung und einen Preis haben.
- Beim Anlegen einer Leistung sollen Beschreibung und Preis festgelegt werden.
- Beschreibung und Preis sollen abgefragt werden können.
- Einer *Rechnung* soll je ein Array für die erbrachten Leistungen und die zugehörigen Beträge haben, welche der Methode *addPosition* übergeben, und dann in die Arrays geschrieben werden sollen.
- Für die Rechnung soll die Gesamtsumme berechnet werden können.
- Instanzvariablen sollen nach außen nicht sichtbar sein.
- Methoden sollen öffentlich aufgerufen werden können.

Ergänzen Sie das Klassendiagramm, damit es die gegebenen Anforderungen erfüllt.

15 Punkte

Hinweis: Geben Sie bei Attributen und Methoden die vollständige UML-Syntax an. Tragen Sie auch das passende Symbol für die Beziehung zwischen Rechnung und Leistung ein. Nutzen Sie dafür den Belegsatz.



b) Entwerfen Sie jeweils den Quelltext in einer Ihnen bekannten Programmiersprache für die Abfrage des Leistungspreises.

2 Punkte

Korrekturrand

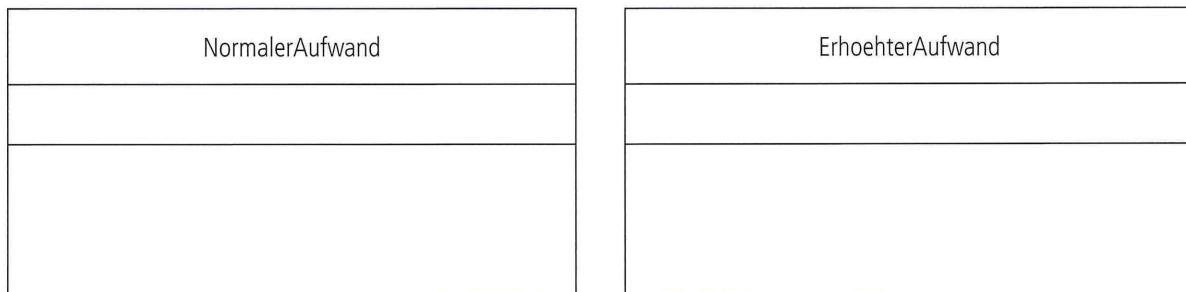
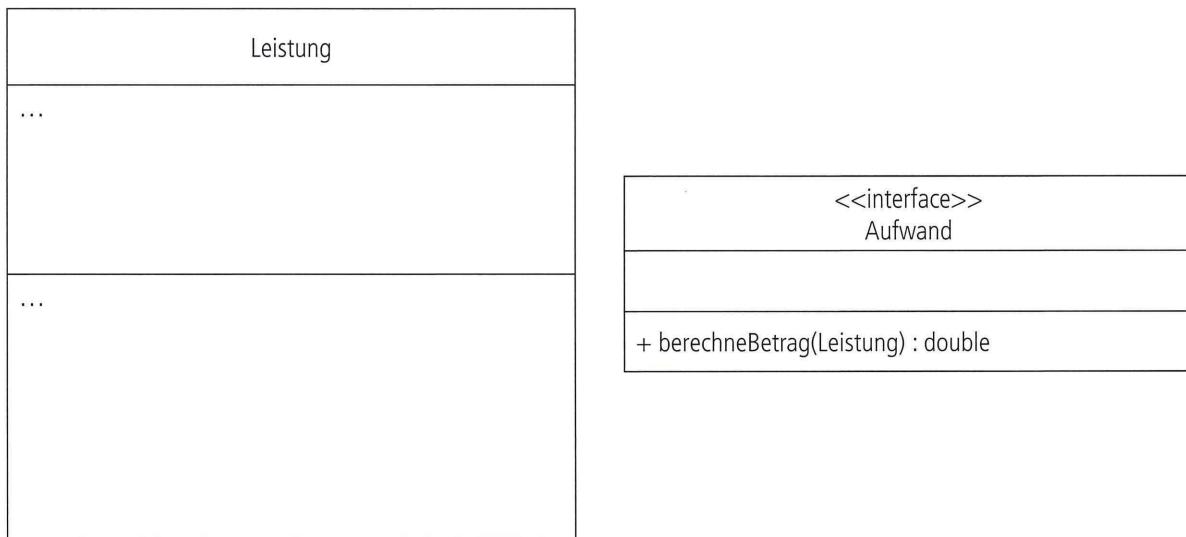
c) Die Beträge für die Leistungen ergeben sich aus dem Preis der Leistung und dem ärztlichen Aufwand. Zur Berechnung des Leistungsbetrages ergeben sich weitere Anforderungen:

- Die Methode *berechneBetrag* soll flexibel eingebunden werden und von der Leistung entkoppelt sein.
- Die Klassen *NormalerAufwand* und *ErhoehterAufwand* realisieren das Interface *Aufwand*.
- Dazu soll die Klasse *Leistung* um eine Referenz *aufwand* vom Typ des Interface *Aufwand*, eine Methode zum Setzen der Aufwandsart und eine Methode *executeBerechneBetrag* zur Ausführung der Betragsberechnung erweitert werden.

Ergänzen Sie das gegebene Klassendiagramm, damit es die gegebenen Anforderungen erfüllt.

5 Punkte

Hinweis: Die Attribute und Methoden der Klasse *Leistung* aus Aufgabe a) müssen nicht wiederholt werden.



d) Entwerfen Sie den Quelltext in einer Ihnen bekannten Programmiersprache für die Methode *executeBerechneBetrag*. 3 Punkte

4. Handlungsschritt (25 Punkte)

Korrekturrand

Das Impfzentrum plant, Informationen über die (Impf-) Produkte, die erforderlichen Bedingungen zur Lagerhaltung, sowie Daten zu Lieferanten und Bestellungen in einer Datenbank zu speichern.

Folgende Bedingungen für das Datenmodell sollen erfüllt sein:

- Jedes Produkt gehört zu einer Produktart
- Jedes Produkt befindet sich an einem Lagerort
- An einem Lagerort sind festgelegte Lagerbedingungen vorhanden
- Jedes Produkt kann von mehreren Lieferanten geliefert werden und jeder Lieferant kann unterschiedliche Produkte liefern
- Eine Bestellung für ein Produkt erfolgt für einen Lieferanten.

a) Erstellen Sie (auf Seite 9) für die beschriebene Situation ein ER-Diagramm. 20 Punkte

b) Erläutern Sie ein Problem, das beim Löschen eines Datensatzes aus der Tabelle Produktart entstehen kann. 2 Punkte

c) Erläutern Sie einen Vorteil, den die referentielle Integrität bei der Umsetzung des Datenmodells in einer relationalen Datenbank bietet. 3 Punkte

5. Handlungsschritt (25 Punkte)

Korrekturrand

Folgende Tabellen (mit Beispieldaten) liegen im Rahmen der Terminierung und Abrechnung vor:

Arzt		
A_Id	A_Nachname	A_Vorname
1	Freudenstedt	Dr. med. Rudolf
2	Nierens	Dr. med. Kirsten
3	Leier	Dr. Patrick
...		

Krankenkasse	
KK_Id	KK_Name
1	TK
2	AOK
3	BKK
4	Knappschaft

Patient							
Pat_Id	Pat_Nachname	Pat_Vorname	Pat_GebDat	Pat_Strasse	Pat_PLZ	Pat_Ort	Pat_KKId
1	Müller	Manni	15.04.1966	Forstweg 12	44456	Musterhausen	1
2	Peters	Peter	12.03.1988				1
3	Fransi	Melanie	13.01.1999				2
4	Kastor	Heinz	14.12.1952				1
5	Krenz	Christina	14.02.1977				2
6	Kreisla	Johann	13.01.1999				3
7	Freie	Ilse	02.05.1955				2
8	König	Ihnes	01.03.2002				1

Termin				
T_Id	T_Termin	T_PatId	T_AId	T_Wahrgenommen
1	01.06.2022 12:10:00	1	1	Ja
2	01.06.2022 12:20:00	2	1	Nein
3	01.06.2022 12:10:00	3	2	Ja
4	01.06.2022 12:40:00	4	1	Ja
5	01.06.2022 12:50:00	6	1	Ja
6	01.06.2022 13:10:00	8	1	Ja
7	01.06.2022 12:20:00	7	1	Ja

- a) Erstellen Sie eine SQL-Abfrage, mit der Sie die Anzahl der Termine im Juni 2022 von jedem hinterlegten Arzt nach folgendem Ergebnisbeispiel erhalten:
4 Punkte

Arzt	Termine
Freudenstedt	6
Nierens	1
Leier	0
...	

- b) Erstellen Sie eine SQL-Abfrage, mit der Sie die Patienten und deren zugehörige Krankenkasse nach folgendem Ergebnisbeispiel erhalten:
 (Sortierung nach KK_Name absteigend und Pat_GebDat aufsteigend)

Korrekturrand

6 Punkte

KK_Name	Pat_Id	Pat_Nachname	Pat_Vorname	Pat_GebDat	Pat_Strasse	Pat_PLZ	Pat_Ort	Pat_KKId
TK	4	Kastor	Heinz	14.12.1952				1
TK	1	Müller	Manni	15.04.1966	Forstweg 12	44456	Musterhausen	1
TK	2	Peters	Peter	12.03.1988				1
TK	8	König	Ihnes	01.03.2002				1
BKK	6	Kreisla	Johann	13.01.1999				3
AOK	7	Freie	Ilse	02.05.1955				2
AOK	5	Krenz	Christina	14.02.1977				2
AOK	3	Fransi	Melanie	13.01.1999				2

- c) Erstellen Sie ein SQL-Abfrage, mit der Sie für alle Krankenkassen eine Abrechnung der Termine im Juni 2022 erhalten.

10 Punkte

Wahrgenommene Termine werden mit 22,50 EUR berechnet, nicht wahrgenommene Termine bleiben unberücksichtigt.

Die Sortierung soll nach Krankenkasse (KK_Name) aufsteigend vorgenommen werden.

Beispielergebnis:

Krankenkasse	Betrag
AOK	45
BKK	22,5
Knappschaft	0
TK	67,5

Fortsetzung 5. Handlungsschritt

Korrekturrand

d) Im Zusammenhang mit SQL-Statements werden häufig Transaktionen verwendet.

Erläutern Sie den Begriff Transaktion.

2 Punkte

e) Beschreiben Sie die Funktion für folgende Syntax im Bezug einer Transaktion

3 Punkte

BEGIN TRANSACTION

COMMIT

ROLLBACK

PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

- 1 Sie hätte kürzer sein können.
- 2 Sie war angemessen.
- 3 Sie hätte länger sein müssen.

Belegsatz

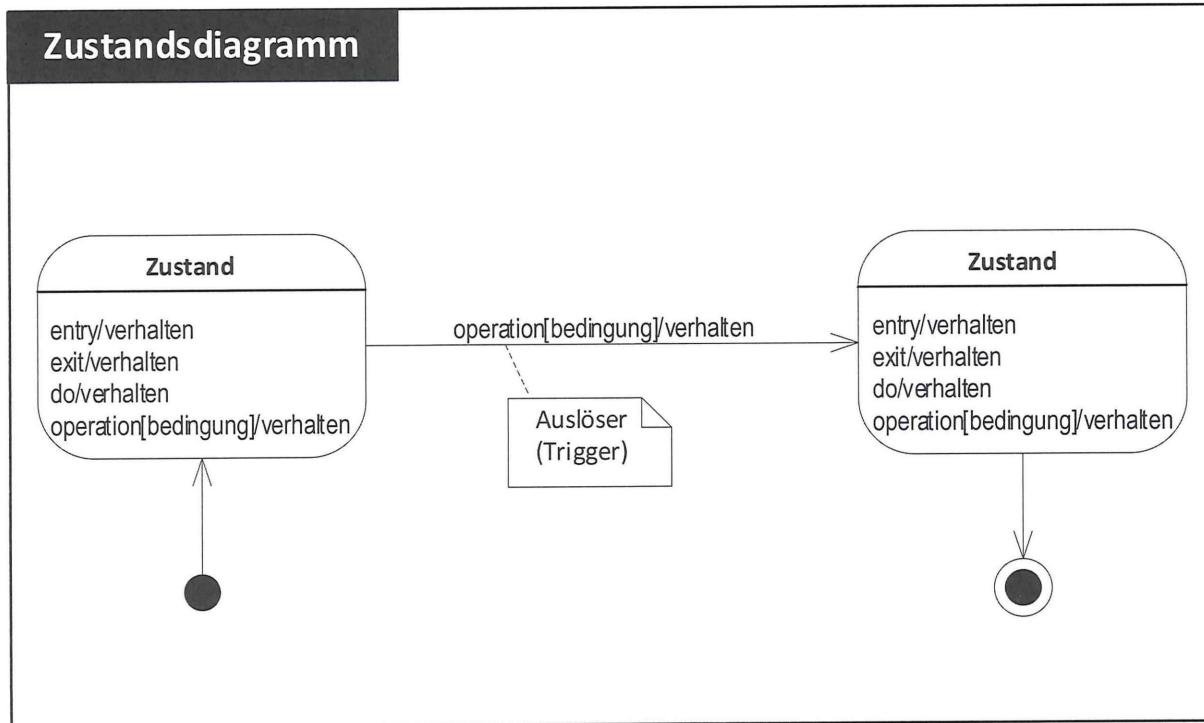
Fachinformatiker Anwendungsentwicklung (AO 1997)
Fachinformatikerin Anwendungsentwicklung (AO 1997)
1196

1 Ganzheitliche Aufgabe I Fachqualifikationen

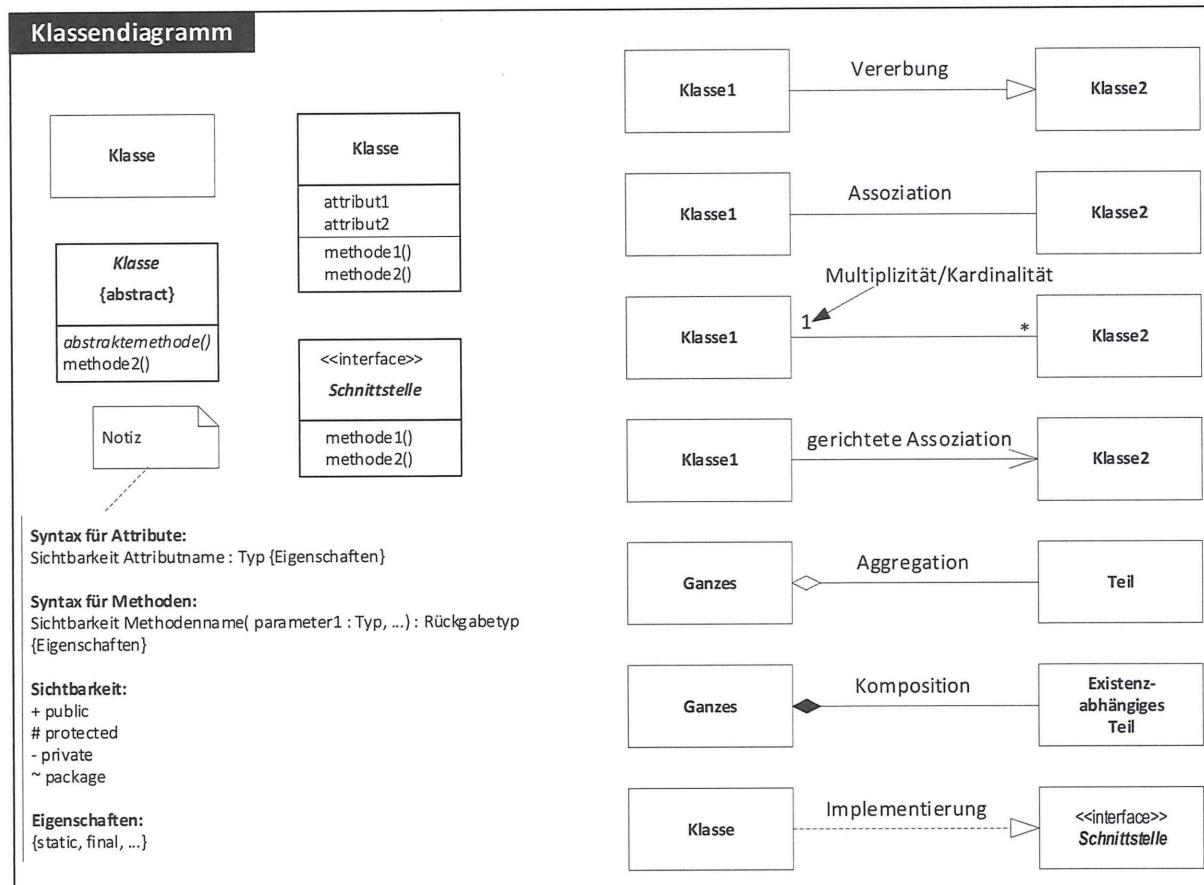
Inhalt

UML-Zustandsdiagramm	2
UML-Klassendiagramm	2
SQL-Syntax (Auszug)	3/4

UML-Zustandsdiagramm



UML-Klassendiagramm



SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname ADD COLUMN Spaltenname Datentyp DROP COLUMN Spaltenname Datentyp ADD FOREIGN KEY (Spaltenname) REFERENCES Tabellenname(Primärschlüsselpaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) REFERENCES Tabellenname(Primärschlüsselpaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... (SELECT ... FROM ... WHERE ...) AS xyz FROM ... WHERE ...	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: xyz) ausgegeben.
SELECT DISTINCT	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery) WHERE NOT EXISTS (subquery)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
GROUP BY Spaltenname1 [, Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [, Spaltenname2, ...] ASC DESC	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
Syntax	Beschreibung
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname SET	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname VALUES (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder SELECT ... FROM ... WHERE	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind

Fortsetzung SQL-Syntax (Auszug) →

SQL-Syntax (Auszug)

Aggregatfunktionen	
AVG(Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen)	Liefert Anzahlzeichen der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen)	Liefert Anzahlzeichen der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert)	Wandelt einen Wert in ein Datum um
DAY(Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR(Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: DAY, MONTH, YEAR	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Textattributen auf Gleichheit, Verwendung von Platzhaltern möglich.
NOT	Logische Negation
OR	Logisches ODER
IS NULL	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2018-03-29