

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, 24. November 2021

Abschlussprüfung Winter 2021/22

1196

1 Ganzheitliche Aufgabe I Fachqualifikationen

5 Handlungsschritte
mit Belegsatz
90 Minuten Prüfungszeit
100 Punkte

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung (AO 1997)

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 programmierte, 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!

Nicht bearbeiteter Handlungsschritt ist Nr.

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			

Prüfungsort, Datum

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:

Korrekturrand

Die WärL Chemie GmbH expandiert und plant in diesem Zusammenhang ein neues Gebäude. Für dieses Projekt übernimmt die IT-Abteilung der WärL Chemie GmbH selbst die Realisierung der Gebäudesteuerung und Wartungsarbeiten.

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

1. Zustandsdiagramm und Anwendungsfalldiagramm für Controller und Lichtsteuerung erstellen
 2. OOP-Methoden für die Auswertung von Temperaturmessungen implementieren
 3. UML-Modellierung für ein Smartphone-Dashboard zur Anzeige von Tankfüllständen anfertigen
 4. ER-Modell zur Speicherung von Sensordaten anlegen
 5. SQL-Abfragen für eine Zeiterfassungsdatenbank formulieren

1. Handlungsschritt (25 Punkte)

- a) Für jeden Flur der Gebäude der WärL Chemie GmbH soll eine automatisierte Lichtsteuerung eingerichtet werden.

Der Controller zur Lichtsteuerung kann sich in einem der Zustände „wartend“, „auto“, „manuell“ oder „zeitgesteuert“ befinden.

- Der Controller befindet sich initial im Zustand „wartend“ bei ausgeschaltetem Licht.
 - Wenn Sensoren Personen im Flur erkennen, wechselt der Controller in den Zustand „auto“ und das Licht wird mit dem Eintritt in diesen Zustand eingeschaltet.
 - Wenn keine Personen mehr im Flur erkannt werden und sich der Controller im Zustand „auto“ befindet, findet ein Wechsel in den Zustand „zeitgesteuert“ statt.
 - Falls innerhalb einer festgelegten Zeitspanne erneut Personen erkannt werden, erfolgt ein Wechsel vom Zustand „zeitgesteuert“ in den Zustand „auto“.
 - Falls innerhalb der Zeitspanne keine Personen den Flur betreten, erfolgt ein Wechsel in den Zustand „wartend“ und das Licht wird bei Eintritt in diesen Zustand ausgeschaltet.
 - Da beim Übergang vom Zustand „wartend“ in den Zustand „auto“ das Licht etwas verzögert eingeschaltet wird, kann eine Person (im Zustand „auto“) das Licht manuell einschalten, solange es noch aus ist. Dann erfolgt ein Zustandswechsel in den Zustand „manuell“ und das Licht wird mit Eintritt in diesen Zustand eingeschaltet.
 - Im Zustand „manuell“ kann das Licht von einer Person ausgeschaltet werden. Der Controller wechselt dabei in den Zustand „wartend“. Falls von den Sensoren keine Personen mehr erfasst werden, erfolgt ein Übergang vom Zustand „manuell“ in den Zustand „zeitgesteuert“.

Erstellen Sie für die beschriebene Situation ein Zustandsdiagramm für den Controller der Lichtsteuerung.

15 Punkte

b) Im Rahmen der neuen Aufgaben für das Gebäudemanagement fallen für die Mitarbeiter folgende Tätigkeiten an:

Korrekturrand

- Für Wartungsmitarbeiter fallen entsprechende Wartungen und Kalibrierungen für die Sensoren an. Für die Wartung ist ein Login erforderlich, falls dieser noch nicht erfolgt ist.
- Administratoren können Sensordaten auswerten. Dazu müssen in jedem Fall Sensordaten gelesen werden. Falls dies noch nicht erfolgt ist, ist für die Auswertung der Daten ein Login erforderlich.
- Jeder Mitarbeiter kann die Sensordaten auslesen.

Erstellen Sie für die beschriebene Situation ein Anwendungsfalldiagramm.

10 Punkte

A large grid of squares, approximately 20 columns by 20 rows, intended for drawing a use case diagram (Anwendungsfalldiagramm).

2. Handlungsschritt (25 Punkte)

Korrekturrand

In einem Gebäudeteil gibt es Sensoren, die Temperaturwerte in unregelmäßigen Zeitabständen messen. Zur Auswertung der Messwerte sollen u. a. zwei Methoden implementiert werden. Folgende Klassen sind bereits vorhanden:

	Value
- sensor_id : Integer	
- value: Double	
- time: Long	
+ Konstruktor(sensor_id: Integer, value: Double, time: Long)	
+ getId() : Integer	
+ getValue() : Double	
+ getTime(): Long	

TempList	
+setValue(value: Value)	Speichert ein Value-Objekt chronologisch in einer Liste. Die Objekte werden für jeden Sensor getrennt gespeichert.
+getValue(sensor_id, pos: Integer) : Value	Liefert für den Sensor mit der übergebenen Sensor-Id das Value-Objekt an der Position pos.
+getSize(sensor_id: Integer): Integer	Liefert die Anzahl der gespeicherten Value-Objekte für den Sensor mit der übergebenen Sensor-Id.

- a) Sobald an einem Sensor eine neue Messung vorliegt, wird automatisch die Methode `onNewValue(sensor_id: Integer, value: Double, time: Long)` aufgerufen.

Die Methode `onNewValue` soll mit folgender Funktionalität implementiert werden:

- Erstellen eines Value-Objektes mit den übergebenen Parametern (siehe Klassendiagramm für Value)
 - Speichern des Value-Objektes mit der Methode setValue des Objektes tempList (das Objekt vom Typ TempList ist bereits vorhanden und kann verwendet werden, siehe Klassendiagramm TempList).

Implementieren Sie die Methode onNewValue in Pseudocode.

5 Punkte

onNewValue(sensor_id: Integer, value: Double, timestamp: Long)

b) Um Temperaturdaten dieses Gebäudes statistisch auswerten zu können, soll eine Methode

Korrekturrand

maxPeriod(sensor_id: Integer, mindestwert: Double): Integer

implementiert werden, die aus allen in tempList gespeicherten Temperaturwerten die höchste Anzahl von hintereinander gespeicherten Werten des Sensors ermittelt, welche den vorgegebenen Mindestwert einhalten.

Beispiel:

Es liegen die Temperaturwerte 20, 22, 23, 21, 19, 18, 20, **22**, **23**, **23**, **24**, **22**, 21 vor.

Die höchste Anzahl von hintereinanderliegenden Werten, welche den Mindestwert 22 einhalten, ist fünf.

Implementieren Sie die Methode maxPeriod in Pseudocode.

20 Punkte

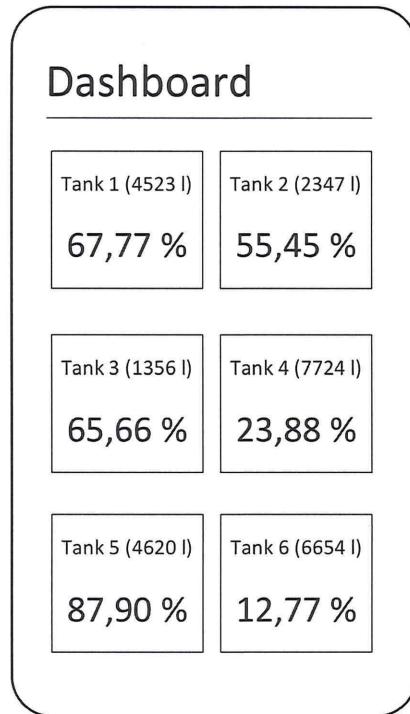
`maxPeriod(sensor_id: Integer, mindestwert: Double): Integer`

Diese Seite bleibt aus drucktechnischen Gründen frei.

3. Handlungsschritt (25 Punkte)

Korrekturrand

Eine Smartphone-Anwendung mit unterschiedlichen Anzeigemöglichkeiten für Tankfüllstände soll entwickelt werden.



In einer ersten Implementierung soll eine „Dashboard“-Anzeige, welche die Füllstände aller Tanks anzeigt und später eine „History“-Anzeige, die den zeitlichen Verlauf des Füllstands eines Tanks darstellt, erstellt werden.

- a) Zunächst soll eine Klasse *Tank* für Tankobjekte modelliert werden.

- aa) Die Klasse *Tank* soll Folgendes beinhalten:

- Die nur klassenintern sichtbaren Instanzvariablen *bezeichner*, *fuellstand*, *fassungsvermoegen*
 - Einen öffentlichen Konstruktor zur Initialisierung der Instanzvariablen.
 - Beispielhaft für den *fuellstand* je eine öffentliche Set- und Get-Methode.

Hinweis: Geben Sie jeweils sinnvolle Datentypen an.

Erstellen Sie das UML-Klassendiagramm für die Klasse *Tank*.

7 Punkte

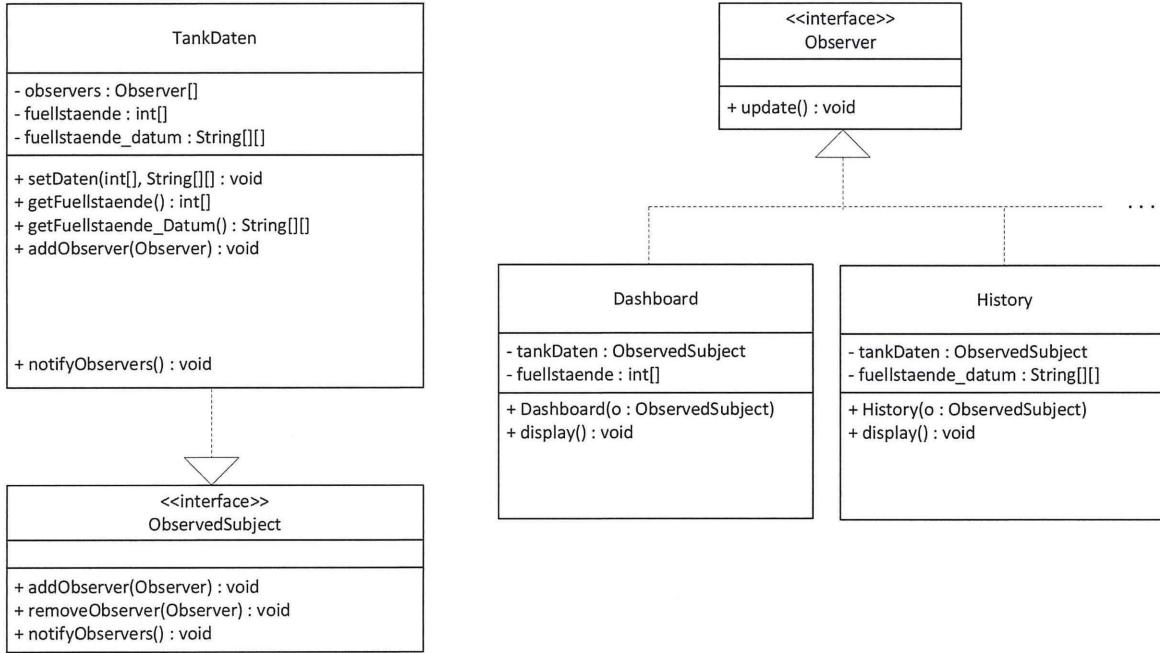
- ab) Implementieren Sie in Pseudocode die Set- und Get-Methode für die Instanzvariable *fullstand* der Klasse *Tank*. 4 Punkte

Fortsetzung 3. Handlungsschritt →

Fortsetzung 3. Handlungsschritt

Korrekturrand

- b) Die Füllstände der Tanks werden stündlich aktualisiert. Alle Anzeigen sollen entsprechend angepasst werden. Ein noch unvollständiger Entwurf nach dem Observer-Muster zur Umsetzung dieser Anforderung liegt bereits vor.



- ba) Ergänzen Sie im obigen Klassendiagramm in den Klassen *TankDaten*, *Dashboard* und *History* die jeweils fehlende Implementierung und die Beziehung zwischen der Klasse *Tankdaten* und dem Interface *Observer*. 4 Punkte
- bb) Erläutern Sie die Art der Beziehung zwischen den Klassen *Tankdaten* und *ObservedSubject*. 2 Punkte
-
-
-
-

c) Zu Dokumentationszwecken soll nachfolgend beschriebener Programmablauf exemplarisch dargestellt werden.

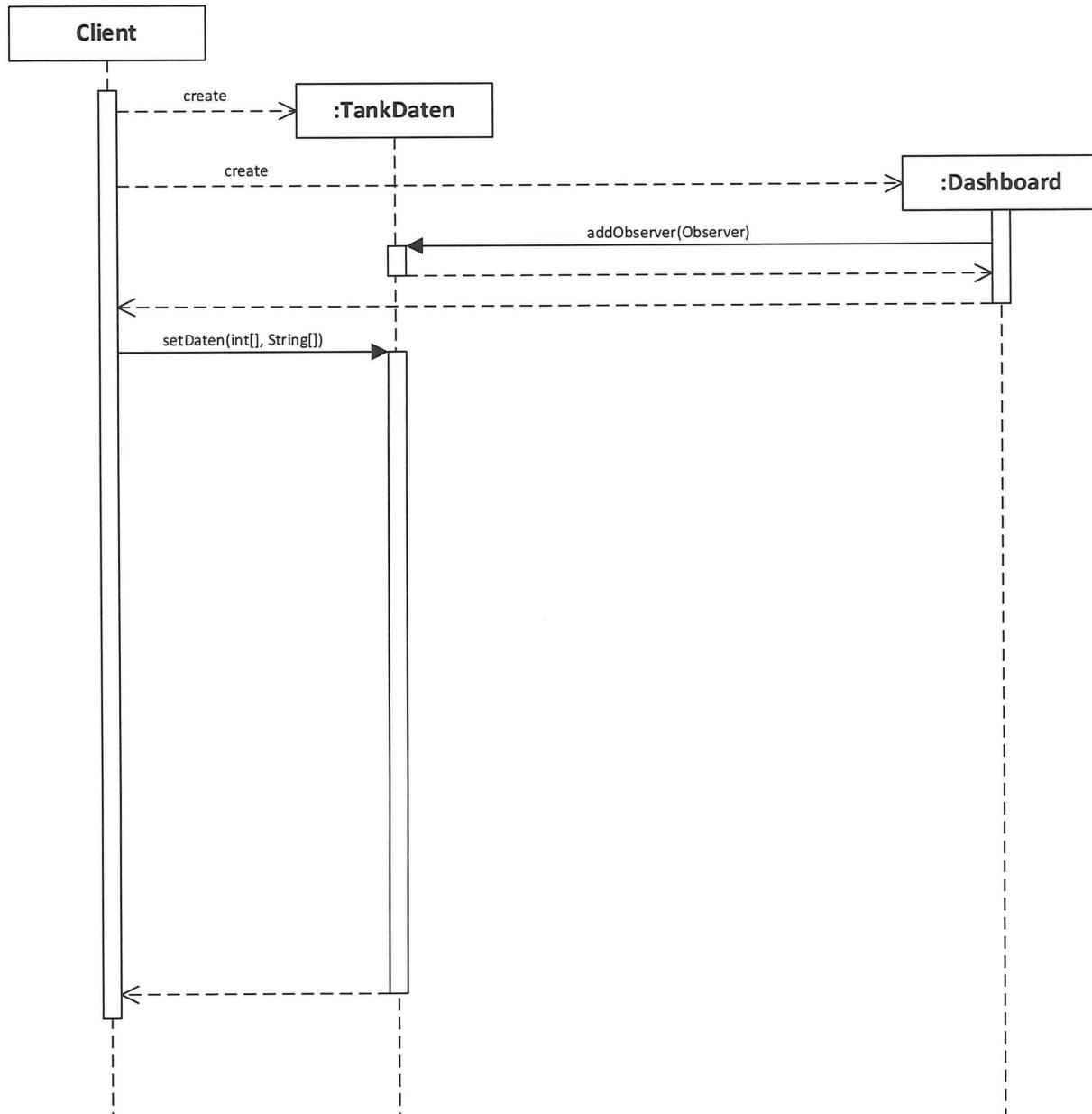
Korrekturrand

ca) Ergänzen Sie das folgende UML-Sequenzdiagramm.

6 Punkte

- Der Client erzeugt ein *TankDaten*- und ein *Dashboard*-Objekt.
 - Im Konstruktor des *Dashboard*-Objekts wird die Methode *addObserver* aufgerufen.
 - Der Client ruft die Methode *setDaten* auf.
 - In der Methode *setDaten* wird *notifyObservers* gestartet.
 - Die Methode *notifyObservers* führt *update* aus.
 - Die Methode *update* holt sich über den Aufruf der Methode *getFuellstaende* das Array mit den aktuellen Füllständen der Tanks und startet die Methode *display* zur Anzeige der Daten.
 - Der Kontrollfluss geht von *display* über *update*, *notifyObservers* und *setData* zurück zum Client.

Hinweis: Die ersten drei Spiegelstriche sind bereits umgesetzt.



cb) Die Methode *notifyObservers* benachrichtigt über den Aufruf von *update* alle registrierten Observer.

Implementieren Sie *notifyObservers* in Pseudocode.

2 Punkte

4. Handlungsschritt (25 Punkte)

Korrekturrand

- a) Die WärL Chemie GmbH soll zur Speicherung der Sensordaten ein ER-Modell erstellen.

 - Jeder Sensor hat eine eigene Sensor-Art.
 - Mehrere Sensoren können von der gleichen Sensor-Art sein.
 - Ein Sensor hat einen festen/bestimmten Standort.
 - An einem Standort können verschiedene Sensoren sein.
 - Von einem Sensor können mehrere Messungen durchgeführt werden.
 - Eine Messung kann verschiedene Aktionen auslösen.
 - Eine Aktion kann von unterschiedlichen Messungen ausgelöst werden.
 - Zu jeder ausgelösten Aktion kann genau ein Eintrag im Aktionsprotokoll gehören.
 - Es gibt mehrere Aktions-Arten, die durch eine Aktion ausgelöst werden können.

Erstellen Sie ein ER-Modell. Attribute müssen nicht erfasst werden.

20 Punkte

- b) Die Tabelle Aktion-Art soll nicht mehr verwendet werden. Daher soll in der Tabelle Aktion das Attribut Art-Id entfernt werden und die Beschreibung der Aktion-Art hinzugefügt werden.

Beschreiben Sie das Problem, welches bei dieser Vorgehensweise auftreten kann.

5 Punkte

5. Handlungsschritt (25 Punkte)

Korrekturrand

Sie sollen SQL-Abfragen für folgende Zeiterfassungsdatenbank erstellen.

Mitarbeiter

MA_ID	Nachname	Vorname	Geb-Datum	Tagesarbeitszeit
811	Müller	Jens	14.04.1995	8
812	Scholz	Birgit	23.08.1977	4
815	Schmidt	Ulrich	02.11.1970	8
817	Storck	Hans	14.11.2003	6
841	Ullmann	Franz	21.12.1972	8
902	Sorge	Susanne	02.03.1965	8

Fehlzeit					
FZ_ID	MA_ID	Von_Datum	Bis_Datum	Grund	Fehltage
1	811	18.10.2021	22.10.2021	Krank	5
2	902	18.10.2021	05.11.2021	Krank	16
3	811	30.12.2021	31.12.2021	Urlaub	2
4	811	03.01.2022	05.01.2022	Urlaub	3
5	815	30.06.2022	30.06.2022	Urlaub	1
6	815	03.07.2022	08.07.2022	Urlaub	6

Hinweis: Jahresübergreifender Urlaub generiert zwei Datensätze (siehe FZ_ID 3 und 4).

- a) Für den Mitarbeiter Ulrich Schmidt wurde in der Tabelle *Fehlzeit* ein Datensatz falsch erfasst:
Statt einer zweitägigen „Dienstreise“ vom 30.06. bis 01.07.2022 wurde versehentlich ein eintägiger „Urlaub“ für den 30.06.2021 eingetragen (siehe Tabelle *Fehlzeit*).

Erstellen Sie eine SQL-Anweisung, mit der die Korrektur durchgeführt werden kann.

5 Punkte

- b) Erstellen Sie eine SQL-Abfrage, welche die Urlaubstage aller Mitarbeiter im Jahr 2021 ermittelt.

10 Punkte

Beispielausgabe:

Bsp. Prüfungsaufgabe:			
811	Müller	Jens	15
812	Scholz	Birgit	10
815	Schmidt	Ulrich	0
817	Storck	Hans	0
841	Ullmann	Franz	21

Fortsetzung 5. Handlungsschritt

Korrekturrand

- c) Die bestehende Datenbank soll wie im Folgenden beschrieben verändert werden.

Erstellen Sie dazu jeweils die SQL-Anweisung.

- ca) Die Tabelle *Fehlzeit* soll gelöscht werden.

2 Punkte

- cb) Es soll eine Tabelle *Fehlzeitgrund* mit folgenden Feldern erstellt werden.

3 Punkte

Fehlzeitgrund

Grund_ID	Grund
1	Urlaub
2	Krank
3	Dienstreise

Hinweis: Es müssen keine Datensätze eingefügt werden.

- cc) Die Tabelle *Fehlzeit* soll in der dargestellten Form neu erstellt werden. In die Tabelle *Fehlzeit* sollen in der Spalte Grund_ID nur solche Werte eingetragen werden können, die in der Tabelle *Fehlzeitgrund* als Primärschlüssel vorkommen. 5 Punkte

Fehlzeit

FZ_ID	MA_ID	Von_Datum	Bis_Datum	Grund_ID	Fehltage
1	811	18.10.2021	22.10.2021	1	5
2	902	18.10.2008	05.05.2021	2	16

PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

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

- Sie hätte kürzer sein können

- 2 Sie war angemessen

- 3** Sie hätte länger sein müssen.



Belegsatz

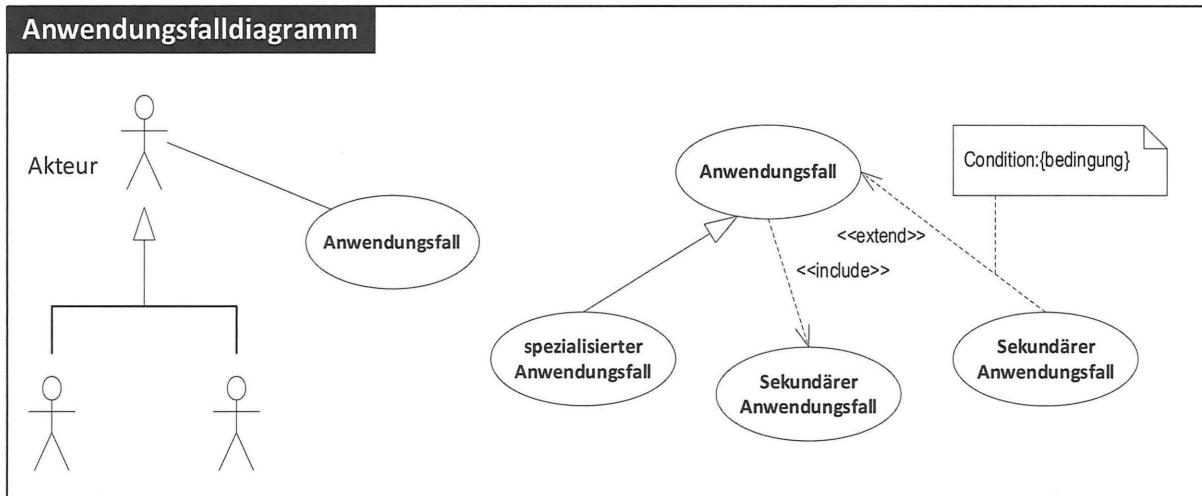
Fachinformatiker Anwendungsentwicklung (AO 1997)
Fachinformatikerin Anwendungsentwicklung (AO 1997)
1196

1 Ganzheitliche Aufgabe I Fachqualifikationen

Inhalt

UML-Anwendungsfalldiagramm	2
UML-Zustandsdiagramm	2
UML-Sequenzdiagramm	3
UML-Klassendiagramm	4
Pseudocode	4
SQL-Syntax (Auszug)	5/6

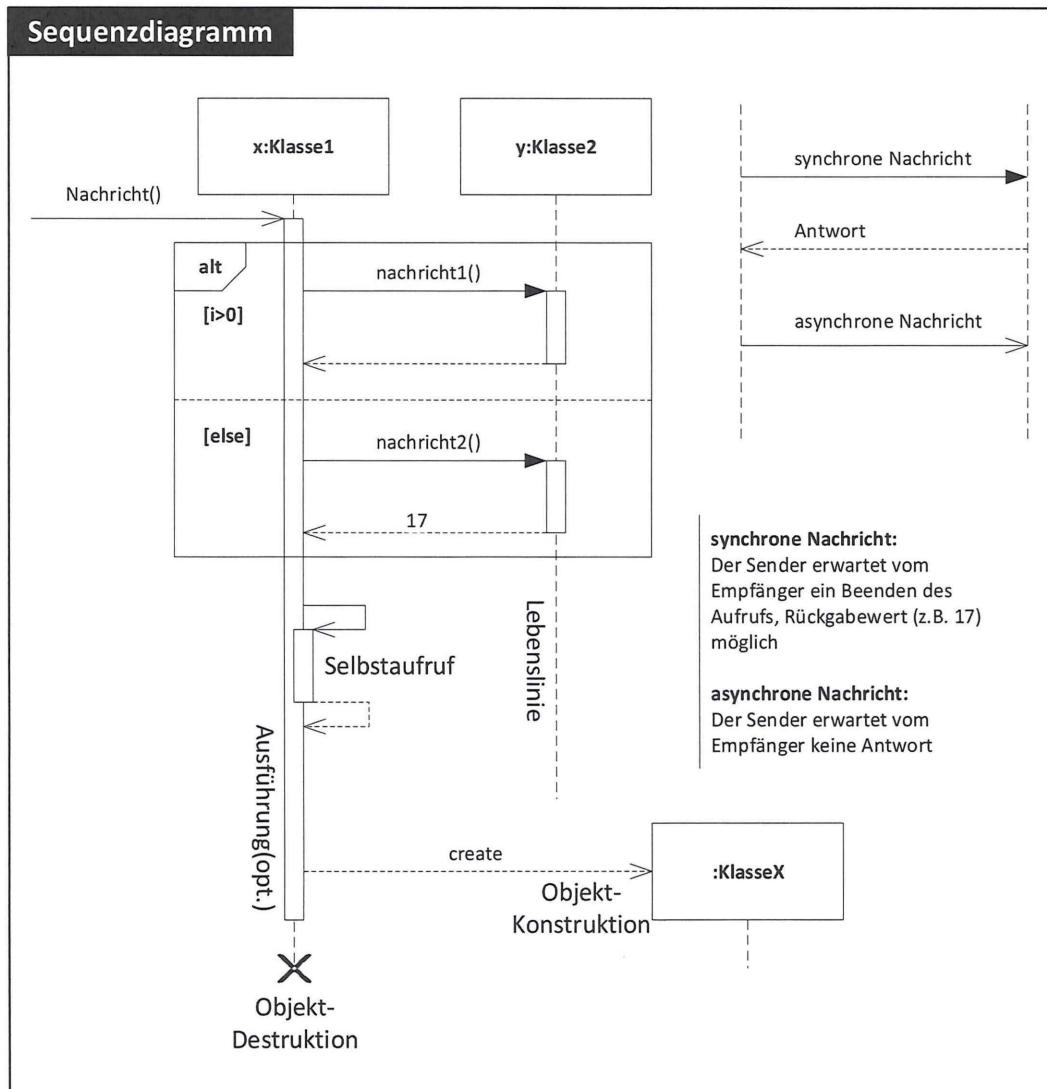
UML-Anwendungsfalldiagramm



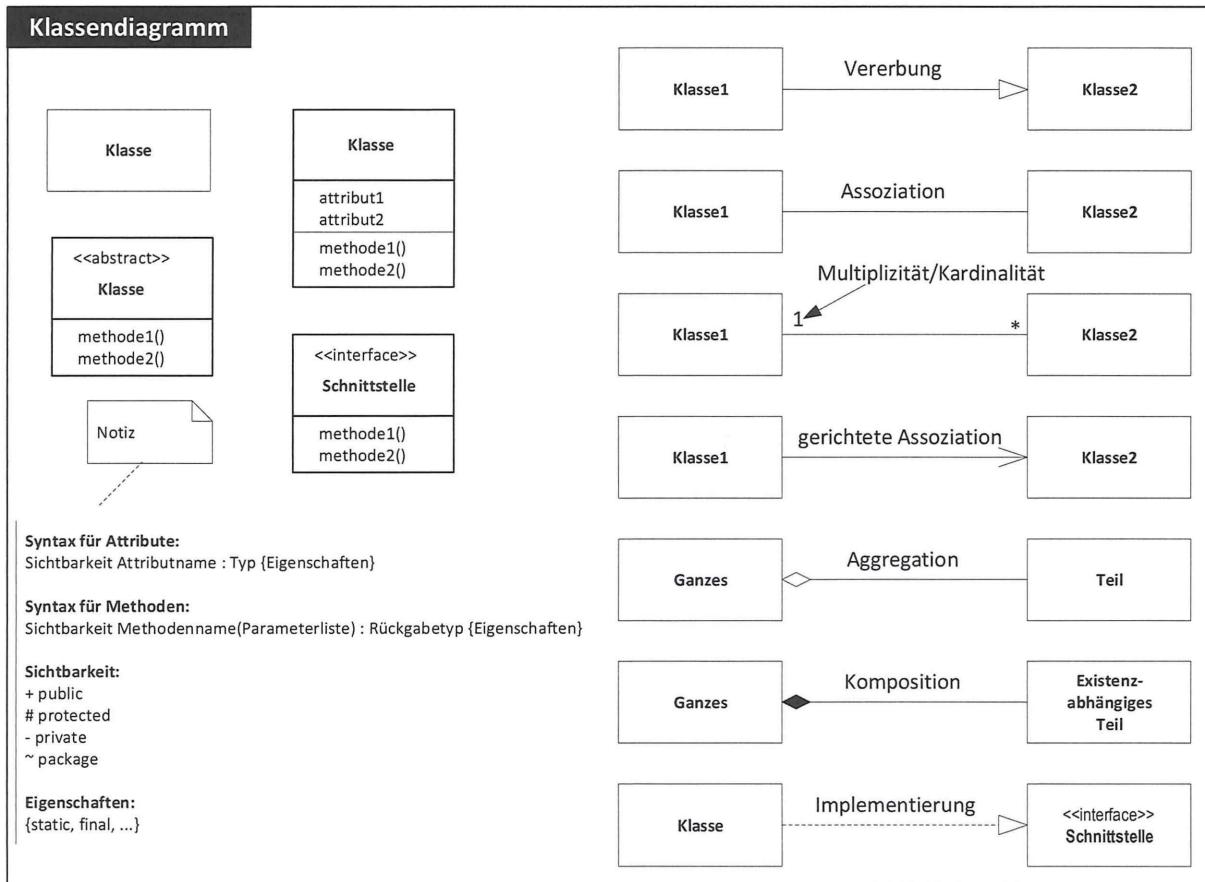
UML-Zustandsdiagramm, Notation (Auszug)

	Anfangszustand
Zustand	Zustand, den ein Element einnehmen kann. z. B. Element = Fenster; Zustände: offen, geschlossen
Beispiel 	Zustandsübergang (Transition) von einem Quellzustand zu einem Zielzustand. In der Beschriftung kann Folgendes angegeben werden: <ul style="list-style-type: none"> - Ein Ereignis (trigger), welches den Zustandsübergang auslöst - Eine Bedingung (guard), welche beim Zustandsübergang erfüllt sein muss - Eine Handlung (action), welche den Zustandsübergang bewirkt
	Selbstaufruf
	Endzustand

UML-Sequenzdiagramm



UML-Klassendiagramm



Pseudocode

Pseudocode	Beschreibung
wenn [Bedingung] dann [Anweisung]	Verzweigung
sonst [Anweisung]	
Ende wenn	
zähle [Variable] von [Startwert] bis [Endwert] [Schleifeninhalt]	Zählschleife (Iteration)
Ende zähle	
solange [Bedingung] [Schleifeninhalt]	Kopfgesteuerte Schleife
Ende solange	
wiederhole [Schleifeninhalt]	Fußgesteuerte Schleife
solange < Bedingung >	
:=	Zuweisung
=; <; >; <=; >=; < >	Vergleiche
integer	
double	
string	
array	

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(Primärschlüsselspaltenname)	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(Primärschlüsselspaltenname)	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	Unterabfrage, die in eine äußere SELECT-Anweisung geschachtelt ist. Das Ergebnis der Unterabfrage wird im Spaltenausdruck (z. B. hier: xyz) ausgegeben.
FROM ...	
WHERE ...	
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)	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefernt. NOT EXIST negiert die Bedingung.
WHERE NOT EXISTS (subquery)	
GROUP BY Spaltenname1 [, Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [, Spaltenname2, ...]	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder
ASC DESC	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 →

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 <i>Anzahlzeichen</i> der Zeichenkette von links.
RIGHT (Zeichenkette, Anzahlzeichen)	Liefert <i>Anzahlzeichen</i> 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	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2018-03-29