

Familienname, Vorname (bitte durch eine Leerspalte trennen)

[illegible]

Berufsnummer

IHK-Nummer

Prüflingsnummer

5	5
---	---

1

--	--

--	--	--

Sp. 1-2

Sp. 3-6

Sp. 7-9

Sp. 10-14

Termin: Mittwoch, 25. November 2015

IHK

1196

1 Ganzheitliche Aufgabe I Fachqualifikationen

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung

5 Handlungsschritte
mit Belegsatz
90 Minuten Prüfungszeit
100 Punkte

Bearbeitungshinweise

1. 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.

2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihre Familiennamen, Ihren Vornamen und Ihre Prüfungs-Nr. in die oben stehenden Felder ein.
3. Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
4. 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.
5. 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.
6. Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
7. Schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
8. Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
10. 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.

Punkte 1. Handlungsschritt 15 16

Punkte 2. Handlungsschritt 17 18

Punkte 3. Handlungsschritt 19 20

Punkte 4. Handlungsschritt 21 22

Punkte 5. Handlungsschritt 23 24

Gesamtpunktzahl 26 27 28

Prüfungsort, Datum

Prüfungszeit

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

Unterschrift _____

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2015 – Alle Rechte vorbehalten!

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

Sie sind Mitarbeiter/Mitarbeiterin der IT-Dienst GmbH, Astadt.

Die IT-Dienst GmbH wurde von der CarStar GmbH mit der Entwicklung eines IT-Systems beauftragt.

Die CarStar GmbH ist ein Autovermieter, der auch Carsharing betreibt.

Sie arbeiten in diesem Projekt mit.

Im Rahmen dieses Projekts sollen Sie vier der folgenden fünf Aufgaben erledigen:

1. Objektorientierte Programmierung vorbereiten und ein UML-Aktivitätsdiagramm erstellen
2. Für eine Software UML-Klassendiagramm, Konstruktor und Methoden entwerfen
3. Eine Methode zur Prüzfziffernberechnung entwerfen
4. Ein Modell für eine relationale Datenbank entwerfen
5. SQL-Anweisungen erstellen

1. Handlungsschritt (25 Punkte)

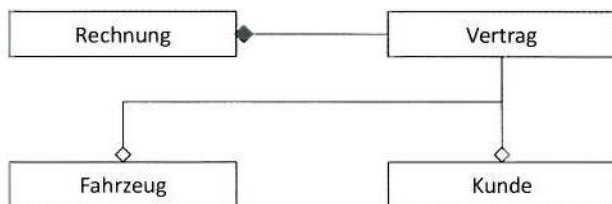
Die IT-Dienst GmbH erstellt eine Soll-Analyse für das durchzuführende Projekt.

Die Software soll mit objektorientierter Programmierung (OOP) realisiert werden.

a) Nennen Sie ein Ziel, das bei der OOP mit Kapselung erreicht werden kann.

4 Punkte

b) Im Rahmen des Projektes wurde folgendes UML-Klassendiagramm erstellt:



Erläutern Sie anhand des gegebenen UML-Klassendiagramms die folgenden Begriffe.

Gehen Sie dabei auf die Lebensdauer der beteiligten Objekte ein.

ba) Aggregation

4 Punkte

bb) Komposition

4 Punkte

- c) Im Zuge der Analyse werden die Klassen Privatkunde und Geschäftskunde definiert.

Korrekturrand

Entwickeln Sie unter Berücksichtigung von Spezialisierung und Generalisierung ein entsprechendes UML-Klassendiagramm.

3 Punkte

Privatkunde	Geschäftskunde
Kunden-ID	Kunden-ID
Vorname	Firma
Name	Straße, Hausnummer
Straße, Hausnummer	PLZ
PLZ	Ort
Ort	Steuernummer
Schufa-Bewertung	Sonderkondition
Geburtsdag	

Hinweis: Notation für UML-Klassendiagramm siehe Belegsatz, Seite 2

Fortsetzung 1. Handlungsschritt →

Fortsetzung 1. Handlungsschritt

Korrekturrand

d) Als Bestandteil der Soll-Analyse soll ein UML-Aktivitätsdiagramm zur Buchung eines Fahrzeuges nach folgenden Vorgaben erstellt werden:

- Der Kunde wählt die gewünschte Anmietstation und den Mietzeitraum aus.
- Das Buchungssystem bietet dem Kunden daraufhin die möglichen Fahrzeugklassen an.
- Der Kunde wählt eine Fahrzeugklasse aus.
- Das Buchungssystem fordert weitere Vertragsdaten an (Name, Adresse usw.).
- Der Kunde gibt die angeforderten Vertragsdaten an.
- Das System verifiziert die Daten des Kunden und sendet eine Vertragsbestätigung und den Zugangscode für das Fahrzeug an den Kunden, falls die Datenprüfung keine Fehler festgestellt hat oder fordert eine erneute Angabe der Vertragsdaten an, falls Fehler festgestellt wurden.

Erstellen Sie das Aktivitätsdiagramm zur Buchung eines Fahrzeuges.

10 Punkte

Hinweis:

- Vorzeitige Abbruchmöglichkeiten durch den Kunden müssen nicht dargestellt werden.
- Die Eingabe von Daten und das Senden dieser Daten können als eine Aktion dargestellt werden.
- Notation für UML-Aktivitätsdiagramm siehe Belegsatz, Seite 3

Kunde	System

2. Handlungsschritt (25 Punkte)

Korrekturrand

Die CarStar GmbH will nun auch Fahrzeuge in Schweden, Norwegen, Island und Dänemark vermieten. Die Rechnungen für dortige Kunden müssen in der jeweiligen Landeswährung ausgestellt werden. – Im bisherigen Abrechnungsverfahren wird ausschließlich in EUR abgerechnet. Dazu werden bislang die vier Klassen *Abrechnung*, *Nutzung*, *Fahrzeug* und *Kunde* verwendet (siehe Klassendiagramm, perforierte Anlage).

Nun soll die bestehende Anwendung erweitert werden, sodass Abrechnungen in den folgenden Fremdwährungen möglich sind. – Dazu soll die Klasse „*AbrechnungExt*“ als Ableitung der Klasse „*Abrechnung*“ entworfen werden.

Staat	Währung	akt_Kurs
Dänemark	DKK	7,461
Island	ISK	147,482
Schweden	SEK	9,222
Norwegen	NOK	8,763

Hinweis:

- Die vier bestehenden Klassen dürfen nicht verändert werden.
- Für die Anwendung darf sich aufgrund von Polymorphie kein Unterschied in der Verwendung der beiden Klassen „*Abrechnung*“ und „*AbrechnungExt*“ ergeben.

Folgende Methode steht in der Klasse *AbrechnungExt* bereits zur Verfügung:

<i>waehrungskurs(staat: String): double</i>	Liefert zur Landeswährung den aktuellen Wechselkurs zum Euro (Default 1,0)
---	--

- a) Erläutern Sie, was in der objektorientierten Programmierung mit Polymorphie bezeichnet wird. 5 Punkte

- b) Sie sollen die Klassen „*Abrechnung*“ und „*AbrechnungExt*“ mit deren Attributen und Methoden sowie deren Beziehung zueinander darstellen.

Entwerfen Sie das entsprechende UML-Klassendiagramm.

5 Punkte

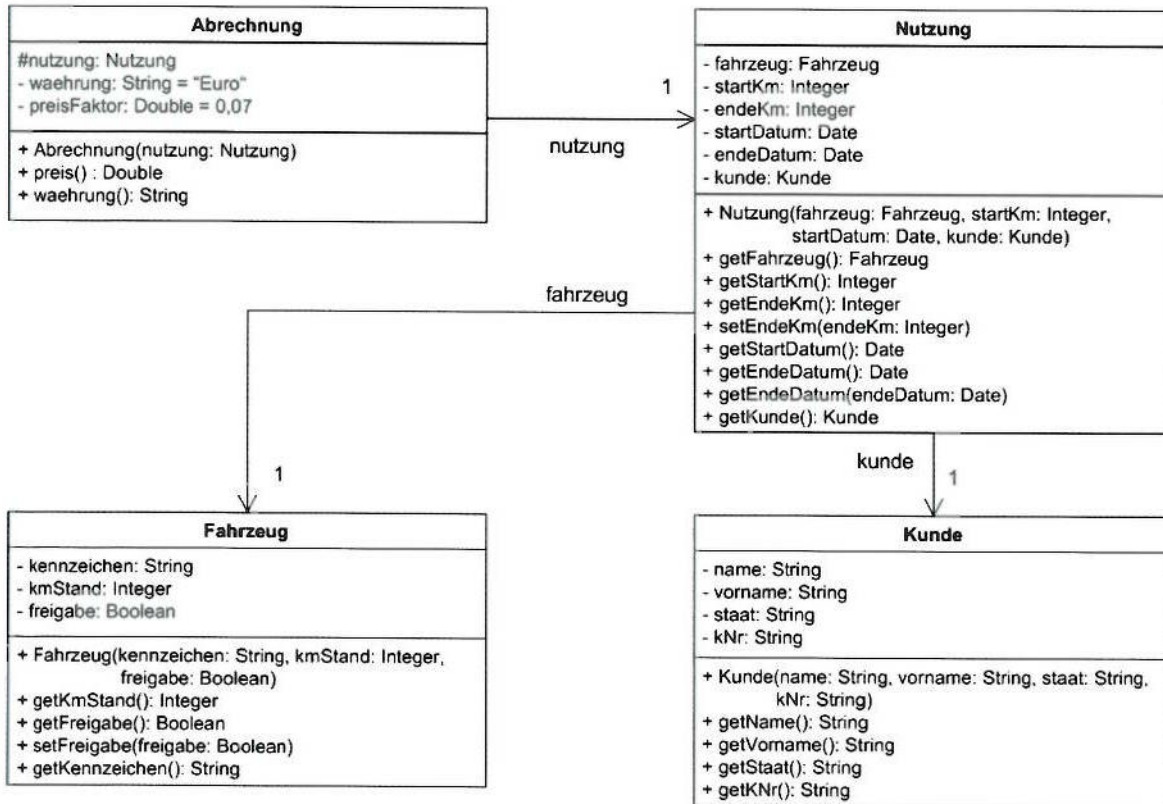
Hinweis: Notation für UML-Klassendiagramm siehe Belegsatz, Seite 2

UML-Klassendiagramm

Abrechnung
#nutzung: Nutzung
- waehrung: String = "Euro"
- preisFaktor: Double = 0,07
+ Abrechnung(nutzung: Nutzung)
+ preis(): Double
+ waehrung(): String

Anlage zum 2. Handlungsschritt

Klassendiagramm



c) Entwerfen Sie jeweils den Code (Pseudocode) für

ca) den Konstruktor *AbrechnungExt*.

3 Punkte

Korrekturrand

cb) die überschriebene Methode *preis()*, die den Preis in der entsprechenden Landeswährung liefern soll; default „Euro“.

6 Punkte

cc) die überschriebene Methode *waehrung()*, die den Währungsstring der entsprechenden Landeswährung liefern soll, default „Euro“.

6 Punkte

3. Handlungsschritt (25 Punkte)

Die IT-Dienst GmbH soll für die CarStar GmbH die Methode *pruefeID* für eine Klasse entwickeln. Die Methode soll ID-Card-Nummern anhand einer Prüfziffernberechnung prüfen.

Beispiel für die Prüzziffernberechnung der ID 6258431979:

	Ziffernstelle									PZ*	Ergebnis
	1	2	3	4	5	6	7	8	9		
ID	6	2	5	8	4	3	1	9	7	9	
Schritt 1	12		10		8		2		14		
Schritt 2	3		1		8		2		5		19
Schritt 3		2		8		3		9			22
Schritt 4	19 + 22										41
Schritt 5	41 auf nächstgrößere durch 10 teilbare Zahl aufrunden										50
Schritt 6	50 – 41										9 (PZ*)

* PZ = Prüfziffer

Der Algorithmus soll nach folgenden Vorgaben entwickelt werden:

Schritt 1: Multiplikation aller Ziffern an ungerader Stelle mit 2.

Schritt 2: Bildung der Quersummen aller entstandenen Produkte und Addition aller entstandenen Quersummen.

Schritt 3: Addition aller Ziffern an gerader Stelle.

Schritt 4: Addition der Ergebnisse aus den Schritten 2 und 3.

Schritt 5: Aufrundung des Ergebnisses aus Schritt 4 auf die nächstgrößere durch 10 teilbare Zahl, falls das Ergebnis nicht ohne Rest durch 10 teilbar ist.

Schritt 6: Berechnung der Differenz aus dem Ergebnis aus Schritt 5 und dem Ergebnis aus Schritt 4.

Die ID-Card-Nummer wird der Methode als String übergeben.

Stimmen die letzte Ziffer der ID-Card-Nummer und die errechnete Prüfziffer überein, gibt die Methode *true*, sonst *false* zurück.

Stellen Sie die Logik der Methode *pruefeID* in Pseudocode, in einem Struktogramm oder PAP dar.

4. Handlungsschritt (25 Punkte)

Korrekturrand

Die monatliche Rechnungserstellung für den Bereich „Carsharing“ soll automatisch erfolgen. Alle zur Rechnungserstellung erforderlichen Daten sollen in einer relationalen Datenbank gespeichert werden.

CarStar GmbH

Gotthilf-von-Meyer-Bogen 12
99900 Scharehausen

CarStar GmbH, Gotthilf-von-Meyer-Bogen 12, 99900 Scharehausen
Firma Klaus Hinundweg
Unterm Flaucher 12c
99912 Ohnau

Kunden-Nr.: K10200
Rechnungsnummer: 102015/005498
Rechnungsdatum: 06.11.2015

Monatsabrechnung: Carsharing Oktober 2015
Abrechnungszeitraum: 01. bis 31. Oktober 2015

Aufstellung Nutzungsdaten, Oktober 2015

Fahrt	Nutzer	Fahrzeug- Typ	Kennzeichen	Abholung Datum/Zeit	Rückgabe Datum/Zeit	Dauer in Std.	EUR/Std.	Std. in EUR	Start km	Ende km	Strecke in km	EUR/km	km in EUR
1	KHG-11	Transporter	SHH-CS 605	01.10./ 09:30	02.10./ 16:36	31,1	5,00	155,00	39.155	39.387	232	0,35	81,20
2	KHG-01	PKW-S	SHH-CS1116	13.10./ 14:15	13.10./ 16:36	2,35	1,90	4,47	5.666	5.715	49	0,19	9,31
3	KHG-11	Kleinbus	SHH-CS 538	14.10./ 16:25	16.10./ 23:36	7,18	3,70	26,57	57.210	57.999	789	0,42	331,38
4	KHG-02	Transporter	SHH-CS 604	21.10./ 17:07	21.10./ 21:36	4,48	5,00	22,40	41.554	41.621	67	0,35	23,45

	EUR
Monatspauschale:	9,20
Nettobetrag km:	445,34
Nettobetrag Std.:	208,44
Gesamt:	662,98
Mehrwertsteuer 19 %:	125,97
Rechnungsbetrag:	788,95

Bitte überweisen Sie innerhalb der gesetzlichen Fristen ohne Abzug.

- a) Sie werden beauftragt, anhand des vorliegenden Auszugs einer Monatsabrechnung ein entsprechendes semantisches Datenmodell (ER-Modell) zu entwickeln. 17 Punkte

Hinweis:

Das ER-Modell soll die Entitäten und deren Beziehungen, jedoch keine Attribute enthalten.

Fortsetzung 4. Handlungsschritt

Korrekturrand

b) Entwerfen Sie für das relationale Datenbankmodell in der dritten Normalform alle erforderlichen Tabellen.

- Geben Sie in jeder Tabelle nur die Schlüsselattribute an.
- Kennzeichnen Sie Primärschlüssel mit PK und Fremdschlüssel mit FK.

Siehe Beispiel Tabelle *Kunde*

8 Punkte

Kunde
Kunden_ID (PK)

5. Handlungsschritt (25 Punkte)

Korrekturrand

Neben der Autovermietung vertreibt die CarStar GmbH Auto-Zubehörteile über einen Shop auf Rechnung.

Folgende Datenstruktur liegt Ihnen für die Auswertungen und zur Bearbeitung vor:

Artikel
Art_ID
Art_Nummer
Art_Bezeichnung
Art_Preis
Art_MwStSatz
Art_Markierung
Art_LfID

Kunde
Kd_ID
Kd_Name
Kd_Vorname
Kd_Strasse
Kd_PLZ
Kd_Ort
Kd_ZbID

Zahlungsbedingung
Zb_ID
Zb_Text
Zb_SkontoFristTage
Zb_SkontoProzent
Zb_GesamtFristTage

Rechnung
Rg_ID
Rg_Nummer
Rg_Datum

Lieferant
Lf_ID
Lf_Firma
Lf_Strasse
Lf_PLZ
Lf_Ort

Sie sind beauftragt worden, aus den vorhandenen Daten die nachfolgend beschriebenen Abfrage-Ergebnisse zu erlangen:

Hinweis: SQL-Syntax (Auszug) siehe Belegsatz, Seiten 4 und 5

- a) Erstellen Sie **eine** SQL-Anweisung, mit der für alle Artikel, welche einen MwSt.-Satz von 7 % aufweisen, das Feld Art_Markierung auf 1 und bei allen anderen Artikeln das Feld Art_Markierung auf 0 gesetzt wird.

4 Punkte

- b) Erstellen Sie eine SQL-Abfrage, mit der alle Kunden wie folgt aufgelistet werden, bei denen eine Zahlungsbedingung mit einem Skontosatz größer 3 % ist, mit Ausgabe der Anzahl aller hinterlegten Rechnungen aus dem Jahr 2015.

AnzRechnung	Kd_Id	Kd_Name	Kd_Vorname	Kd_Strasse	Kd_PLZ	Kd_Ort
4	K006423	Mustermann	Manni	Freidorfstr. 44	44894	Neuss
2	K002351	Peters	Uschi	Utrechter Str. 56	54453	Petersdorf

6 Punkte

Fortsetzung 5. Handlungsschritt →

Fortsetzung 5. Handlungsschritt

Korrekturrand

- c) Erstellen Sie eine SQL-Abfrage, mit der alle Kunden mit allen Attributen aufgelistet werden, bei denen im Nachnamen die Zeichenfolge ,oe', ,ö' oder ,õe' enthalten ist. Case-Sensitive ist nicht aktiv. 4 Punkte

- d) Erstellen Sie eine SQL-Abfrage, mit der alle Orte und die Anzahl der Kunden aus dem jeweiligen Ort ausgegeben werden. Die Sortierung soll nach Anzahl der vorhandenen Kunden innerhalb des aufgelisteten Ortes absteigend erfolgen. 4 Punkte

Beispiel

AnzKunden	Ort
21	Petersdorf
19	Moers
14	Neuss
11	Olpe

- e) Erstellen Sie eine SQL-Abfrage, mit der alle Kunden und Lieferanten in einer Tabelle wie folgt aufgelistet werden. Die Sortierung soll nach Name absteigend erfolgen. 7 Punkte

Beispiel

Tabellenname	ID	Name	PLZ	Ort
Kunde	K005423	Ehrlich	55675	Olpe
Lieferant	L000158	Dart-PC GmbH	45955	Mönchengladbach
Kunde	K002148	Coellen	44885	Moers
Kunde	K006767	Benning	44894	Neuss
Lieferant	L000017	Albers GmbH	54453	Petersdorf
...				

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/Fachinformatikerin
Anwendungsentwicklung
1196

1

Ganzheitliche Aufgabe I Fachqualifikationen

UML-Klassendiagramm, Notation

Seite 2

UML-Aktivitätsdiagramm, Notation

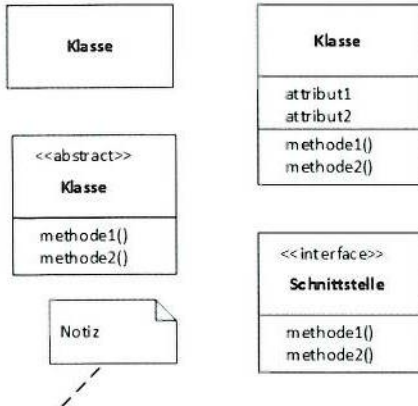
Seite 3

SQL-Syntax (Auszug)

Seite 4-5

UML-Klassendiagramm, Notation

für 1. Handlungsschritt, Teilaufgabe c)
und 2. Handlungsschritt, Teilaufgabe b)



Syntax für Attribute:

Sichtbarkeit Attributname : Typ {Eigenschaften}

Syntax für Methoden:

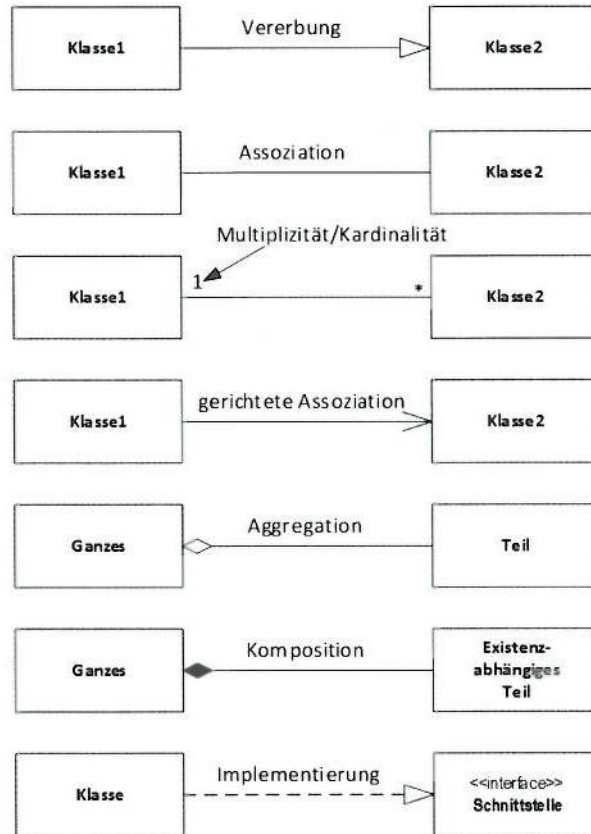
Sichtbarkeit Methodenname(Parameterliste) : Rückgabotyp {Eigenschaften}

Sichtbarkeit:

+ public
protected
- private
~ package

Eigenschaften:

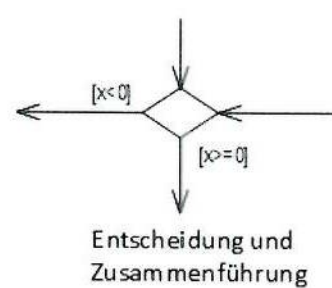
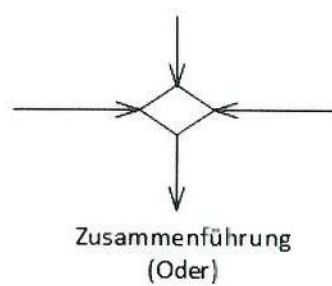
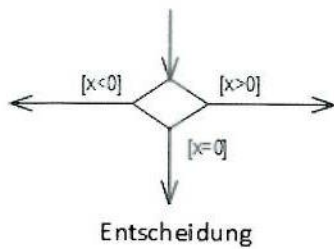
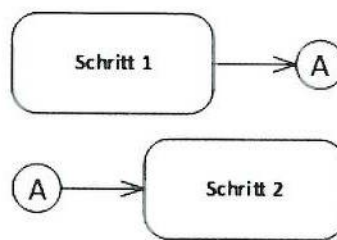
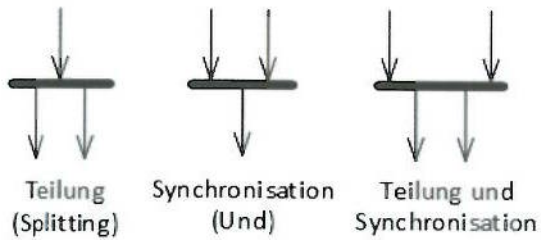
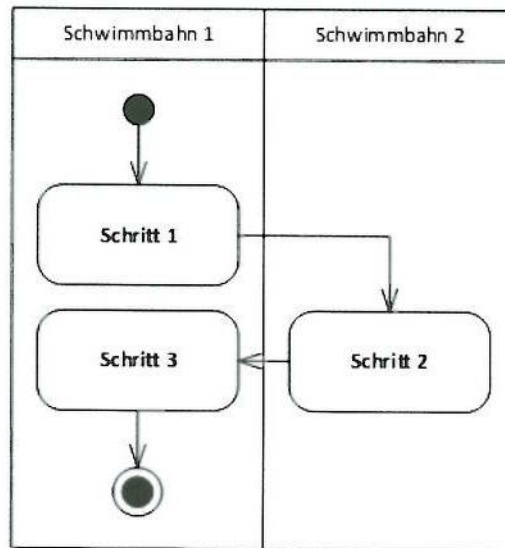
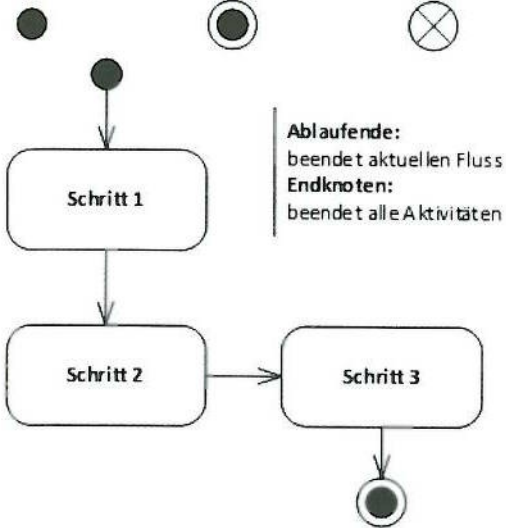
{static, final, ...}



UML-Aktivitätsdiagramm, Notation

für 1. Handlungsschritt, Teilaufgabe d)

Startknoten Endknoten Ablaufende



SQL-Syntax (Auszug)

für 5. Handlungsschritt

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellennamen(Feldname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellennamen ADD COLUMN Spaltenname Datentyp DROP COLUMN Spaltenname	Änderungen in eine Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY	
FOREIGN KEY (Feldname) REFERENCES	
DROP TABLE Tabellennamen	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * Feldname1 [, Feldname2, ...]	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.
CASE WHEN / THEN / ELSE / END	Fallunterscheidung – liefert als Abfrage in Abhängigkeit gebrachte Klartexte für abgefragte Werte (Anzuwenden wie IF / THEN / ELSE / END)
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
GROUP BY Feldname1 [,Feldname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Feldname1 [,Feldname2, ...] ASC DESC	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend
Datenmanipulation	
DELETE FROM Tabellennamen	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellennamen SET	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellennamen ... 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
Aggregatfunktionen	
AVG(Feldname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Feldname *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Feldname Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Feldname Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Feldname Formel)	Ermittelt den größten aller Werte im angegebenen Feld

SQL-Syntax (Auszug) – Fortsetzung

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
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
NOT	Logische Negation
OR	Logisches ODER
=	Test auf Gleichheit
>, >=, <, <=, <>	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen