

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 2014 – Alle Rechte vorbehalten!

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

Sie sind Mitarbeiter/-in der dLine AG, Astadt.

Die dLine AG will ihre Geschäftsprozesse mithilfe neuer IT-Anwendungen optimieren.

Dazu wird ein Projekt durchgeführt, in dem Sie mitarbeiten.

Sie sollen vier der folgenden fünf Aufgaben erledigen:

1. Ein Teilprojekt mit einem Netzplan planen und Cloud Computing erläutern
2. Ein UML-Aktivitätsdiagramm aus einer Ereignisgesteuerten Prozesskette (EPK) ableiten
3. Ein Programm zur Erstellung von Rechnungen entwerfen
4. Ein Programm zur Erstellung von Quick Response Codes (QR-Code) entwerfen
5. SQL-Anweisungen erstellen

1. Handlungsschritt (25 Punkte)

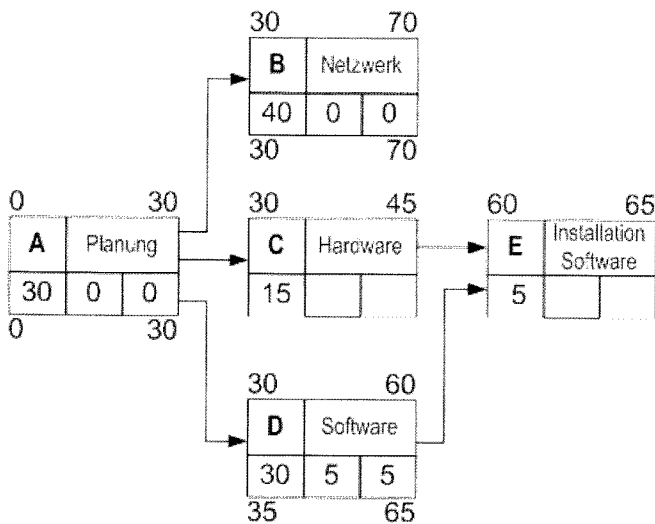
Die Projektgruppe der dLine AG plant das Teilprojekt „Restrukturierung des IT-Systems“ und diskutiert die Nutzung von Cloud Computing.

a) Zum Teilprojekt „Restrukturierung des IT-Systems“ liegt folgende Vorgangsliste vor.

Vorgang	Beschreibung	Tage	Vorgänger
A	Planung	30	-
B	Netzwerk (Erstellung)	40	A
C	Hardware (Bereitstellung)	15	A
D	Software (Bereitstellung)	30	A
E	Installation Software	5	C, D
F	Anschluss Computer	5	B, E
G	Test und Übergabe	10	F

aa) Vervollständigen Sie anhand der Vorgangsliste folgenden Netzplan und ermitteln Sie den kritischen Weg.

10 Punkte



FAZ	FEZ
Vorgang	Beschreibung
Dauer	GP FP
SAZ	SEZ

FAZ = frühester Anfangszeitpunkt
 FEZ = frühester Endzeitpunkt
 SAZ = spätester Anfangszeitpunkt
 SEZ = spätester Endzeitpunkt
 GP = Gesamtpuffer = SAZ – FAZ
 FP = Freier Puffer = FAZ (Nachfolger) - FEZ

Juni 2014						
Mo	Di	Mi	Do	Fr	Sa	So
						1
2	3	4	5	6	7	8*
9*	10	11	12	13	14	15
16	17	18	19*	20	21	22
23	24	25	26	27	28	29
30						

Juli 2014						
Mo	Di	Mi	Do	Fr	Sa	So
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

August 2014						
Mo	Di	Mi	Do	Fr	Sa	So
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

* Pfingsten: 08. und 09. Juni 2014, Fronleichnam: 19. Juni 2014

ab) Mit dem Projekt wird am Montag, 2. Juni 2014, begonnen, samstags, sonntags und feiertags wird nicht gearbeitet.

Korrekturrand

Ermitteln Sie das Datum des Tages, an dem mit dem Vorgang D *Software* frühestens begonnen werden kann. 3 Punkte

ac) Im Verlauf einer Projektgruppensitzung werden Sie aufgefordert die Begriffe Gesamtpuffer (GP) und freier Puffer (FP) zu erklären.

Erläutern Sie Gesamtpuffer und freier Puffer.

4 Punkte

Gesamtpuffer:

Freier Puffer:

b) Die dLine AG will Cloud Computing (CC) nutzen.

ba) Erläutern Sie Cloud Computing (CC).

2 Punkte

bb) Erläutern Sie zwei betriebswirtschaftliche Vorteile, die sich aufgrund der Nutzung von CC ergeben können.

4 Punkte

bc) Im Rahmen von CC sollen Wireless Thin Clients eingeführt werden.

2 Punkte

Erläutern Sie Wireless Thin Client.

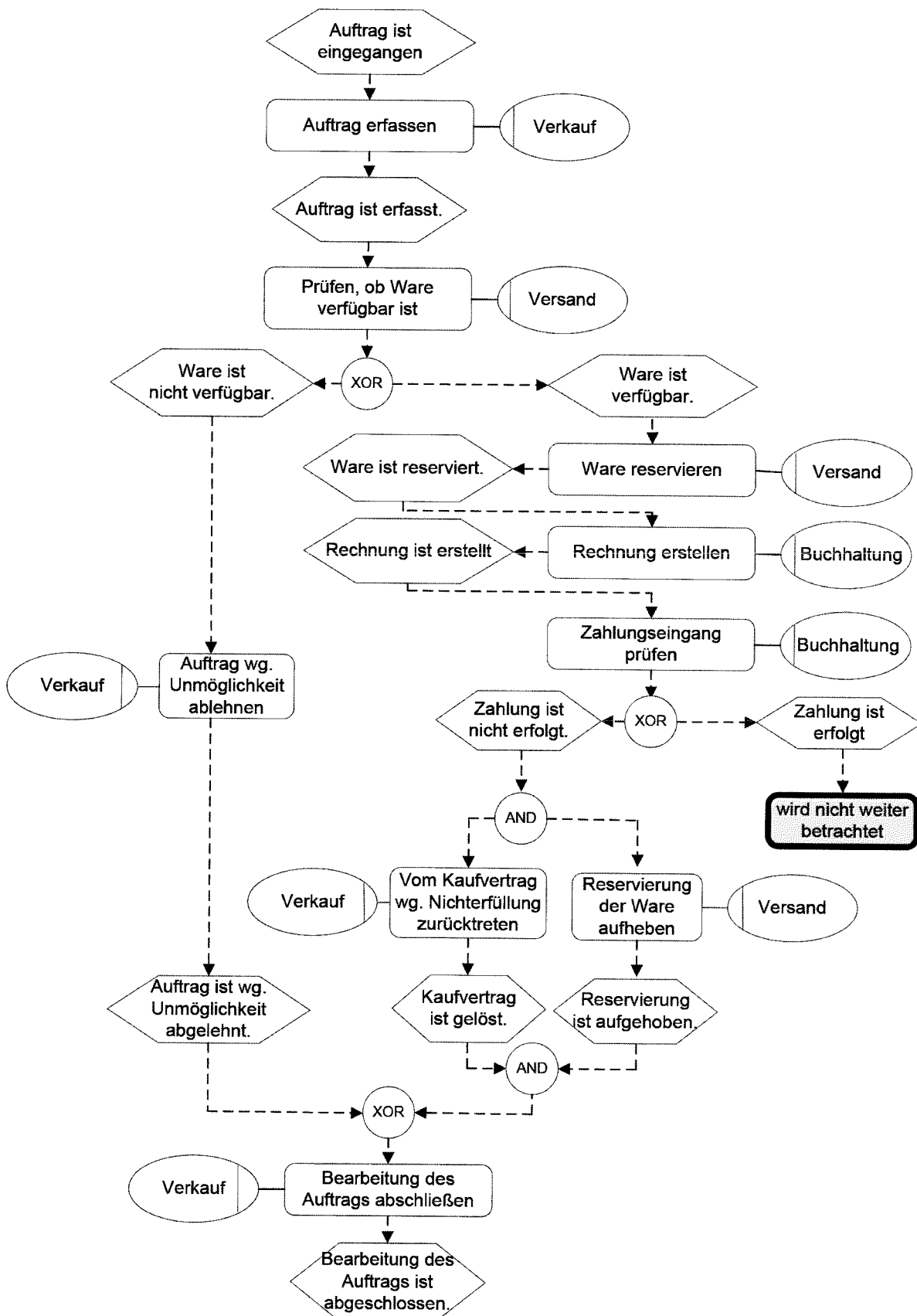
Wichtig!

2. Handlungsschritt (25 Punkte)



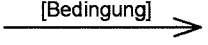
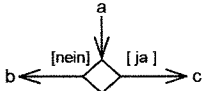
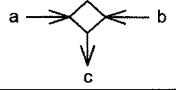
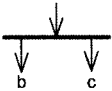
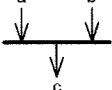


Korrekturrand

Der Informatikkaufmann der dLine AG hat die Bearbeitung eines Auftrags in folgendem EPK dargestellt.

Stellen Sie diesen Ablauf in einem UML-Aktionsdiagramm/Aktivitätsdiagramm auf Seite 7 dar.



UML-Aktionsdiagramm, Notation

Symbol	Beschreibung
	Startknoten: Markiert den Beginn eines Ablaufs
	Aktion: Legt Verhalten fest, das eine Veränderung herbeiführt
	Pfeil: Beschreibt den Fluss zwischen den Aktionen In einer eckigen Klammer kann eine Bedingung angegeben werden.
	Entscheidung: Nach dem Ende der Aktion a wird entweder die Aktion b oder die Aktion c ausgeführt.
	Zusammenführung: Nach dem Ende der Aktionen a oder b wird die Aktion c ausgeführt.
	Splitting: Nach dem Ende einer Aktion werden mehrere Aktionen gleichzeitig begonnen. Z. B.: wenn die Aktion a beendet wurde, werden die Aktionen b und c begonnen.
	Synchronisation: Nach dem Ende mehrerer Aktionen wird eine Aktion ausgeführt. Z. B.: Wenn die Aktionen a und b beendet wurden, wird die Aktion c ausgeführt.
	Ablaufende: Markiert das Ende eines Zweiges
	Endknoten: Markiert das Ende eines Ablaufs

Verkauf	Buchhaltung	Versand

3. Handlungsschritt (25 Punkte)

Korrekturrand

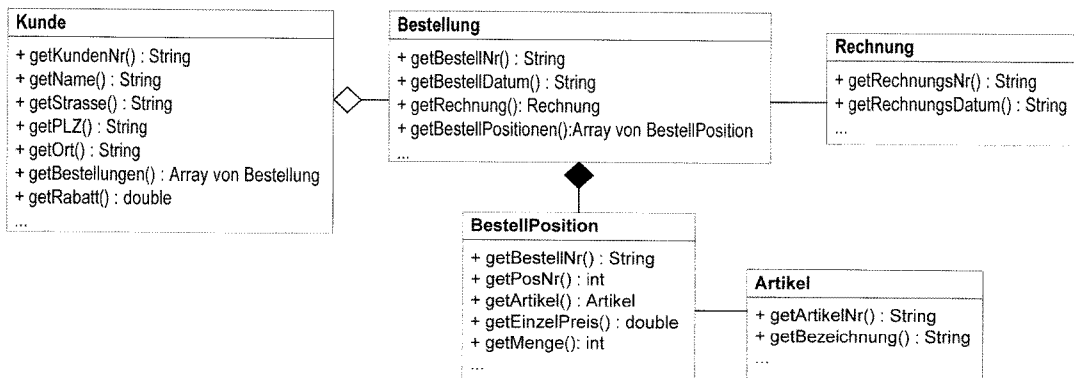
Zur Rechnungserstellung benötigt die dLine AG ein neues Programm.

Bei der zu erstellenden Methode *rechnungsausgabe* sind folgende Angaben zu berücksichtigen:

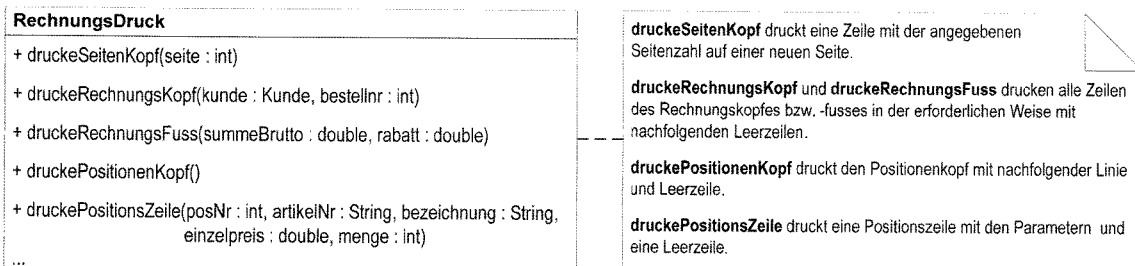
- Die Methode soll für einen Kunden die Rechnungen für dessen Bestellungen ausgeben.
- Der Methode wird eine Referenz auf ein Kundenobjekt als Parameter übergeben.
- Die Methode soll die Rechnungsbeträge der Rechnungen zu einem Gesamtbetrag addieren und zurückgeben.
- Die Methode soll zur Klasse *RechnungsDruck* (siehe unten) gehören.
- Die Methode soll auf einer Rechnungsseite höchstens 20 Positionszeilen ausgeben, danach soll sie einen Wechsel auf die nächste Seite durchführen.
- Jede Rechnung beginnt mit der Seiten-Nummer 1.

Stellen Sie die Methode *rechnungsausgabe(kunde : Kunde) : double* in Pseudocode, in einem Struktogramm oder einem Programmablaufplan (PAP) dar.

Klassen und deren Beziehungen



Klasse *RechnungsDruck*



Beispiel für eine Rechnung (mit Erläuterungen 1 bis 5)

1	Seite 1						Erläuterungen
2	Erwin Mustermann Musterweg 7 77777 Musterstadt						1 = Seitenkopf
	Rechnungsdatum: 14.04.2014 Rechnungsnummer: R1234 Kundennummer: K1234						2 = Rechnungskopf
	Ihre Bestellung B1234 vom 13.03.2014						3 = Positionenkopf
3	Position	ArtikelNr	Bezeichnung	Einzelpreis (EUR)	Menge	Preis (EUR)	4 = Positionszeile
4	1	A1001	Shampoo Haarfein	5.00	10	50.00	5 = Rechnungsfuß
4	2	A0345	Körperlotion M100	3.50	2	7.00	
4	3	A4013	Luxuseife Ramona	25.00	1	25.00	
1	Seite 2						
3	Position	ArtikelNr	Bezeichnung	Einzelpreis (EUR)	Menge	Preis (EUR)	
4	19	A5071	Deodorant. Brise	7.50	3	22.50	
4	20	A1021	Rasierapparat R123	45.00	1	45.00	
Summe netto						211.00	
Rabatt (2%)						- 4.22	
Summe netto abzgl. Rabatt						206.78	
USt (19%)						39.29	
Summe brutto						246.07	

rechnungsausgabe(kunde : Kunde) : double

Korrekturrand

4. Handlungsschritt (25 Punkte)

Korrekturrand

Die dLine AG will QR-Codes mit Werbebotschaften einsetzen. Zur Erstellung dieser QR-Codes soll eine Methode entwickelt werden.

Der Werbetext soll wie folgt codiert werden:

1. Setzen der Kennung „0100“ für den Zeichensatz ISO-8859-1 an den Anfang des Codes
2. Codierung des Textes: Für jedes Zeichen des Werbetextes wird aus dem Zeichensatz ISO-8859-1 der entsprechende Binärcode als String ermittelt und an den Code angefügt.
Der Binärcode setzt sich aus einem Zeilenwert und einem Spaltenwert zusammen (siehe unten stehende Tabelle).
Beispiel: Das Zeichen „G“ hat den Binärcode „01000111“ (Zeile: „0100“, Spalte: „0111“)
3. Anfügen der Ende-Kennung „0000“ an das Ende des Codes

Beispiel: Code für den Werbetext „Günstig!“.

Kennung für Zeichensatz	G	ü	n	s	t	i	g	!	Ende-Kennung
0100	01000111	11111100	01101110	01110011	01110100	01101001	01100111	00100001	0000

Die Zeichen des Zeichensatzes ISO-8859-1 und dessen Binärwerte sind in dem zweidimensionalen String-Array *zeichenSatz* gespeichert. Das Array ist wie folgt aufgebaut (Zeilen- und Spaltenangaben sind nicht Teil des Arrays.):

Array *zeichenSatz*

	Spalte																
Zeile	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	14	16
0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
1	0010	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
2	0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
3	0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
4	0101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
5	0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
6	0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
7	1010	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	—
8	1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
9	1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
10	1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
11	1110	à	á	â	Ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
12	1111	ð	ñ	ò	Ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Folgende Funktionen stehen zur Verfügung

<code>laengeZeichenkette(zeichenkette: String): int</code>	Ermittelt die Länge der Zeichenkette
<code>zeichen(zeichenkette: String, stelle: int): char</code>	Ermittelt das Zeichen in einer Zeichenkette an der angegebenen Stelle

Stellen Sie den entsprechenden Algorithmus für die Methode *qrCode(werbetext: String): String* in Pseudocode, in einem Struktogramm oder einem Programmablaufplan (PAP) dar.

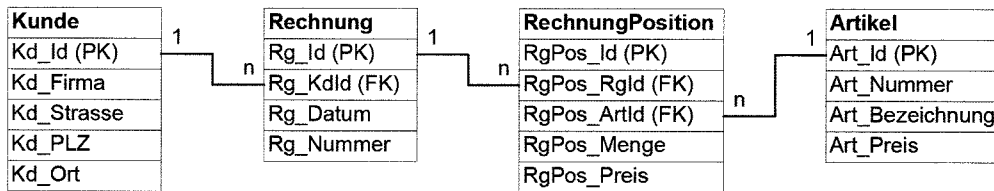
qrCode(werbetext: String): String

Korrekturrand

5. Handlungsschritt (25 Punkte)

Korrekturrand

Die dLine AG verwendet folgendes Datenmodell.



- a) Erstellen Sie eine SQL-Abfrage, die alle Artikel auflistet, deren Artikelbezeichnungen die Zeichenketten „Schmerzmittel“ oder „schmerzmittel“ enthalten. Zu jedem Artikel sollen jeweils alle Attribute ausgegeben werden. 3 Punkte

- b) Erstellen Sie eine SQL-Abfrage, die alle Kunden und deren Umsätze auflistet. Zu jedem Kunden sollen alle Attribute ausgegeben werden. Die Liste soll nach Umsatz absteigend sortiert werden. 7 Punkte

- c) Erstellen Sie eine SQL-Abfrage, die für jeden Artikel Folgendes ermittelt:
- Die Menge, die insgesamt verkauft wurde
 - Die Anzahl der Rechnungspositionen

8 Punkte

Beispielausgabe

Art_Id	Art_Nummer	Art_Bezeichnung	Art_Preis	MengeGesamt	AnzahlRechPos
76723	IB767126	Bosko Schmerzmittel	10,50	451	391
98239	HM67C7	Lidor Kamillentee	3,20	7.932	6.451
09239	87876234-IK	Schnarr Rachendrops	4,00	2.351	2.120

Hinweis: Die Kopfzeile wird nicht mit ausgegeben.

Dieses Blatt kann an der Perforation aus dem Aufgabensatz herausgetrennt werden!

SQL-Syntax-Referenz (Auszug)

Syntax	Beschreibung
Tabellen	
CREATE TABLE Tabellennamen(Feldname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
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
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 Beispiel: FROM Verkäufer LEFT JOIN Kunde ON Verkäufer.Ver_ID = Kunde.Ver_ID
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 Beispiel: FROM Verkäufer RIGHT JOIN Kunde ON Verkäufer.Ver_ID = Kunde.Ver_ID
FULL JOIN	Liefert aus beiden Tabellen jeweils alle Datensätze
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen Beispiel: WHERE name = 'Maier'
GROUP BY Feldname1 [,Feldname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes Beispiel: GROUP BY name, vorname
ORDER BY Feldname1 [,Feldname2, ...] ASC DESC	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend Beispiel: ORDER BY name ASC
Datenmanipulation	
DELETE FROM Tabellennamen	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellennamen SET	Aktualisiert Daten in Feldern einer Tabelle Beispiel: UPDATE Artikel SET(Preis=10.00)
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 Beispiele: INSERT INTO kunde VALUES 56532, 'Martina', 'Schmitz', '12345', 'Berlin' INSERT INTO kunde SELECT * FROM vertrag WHERE stadt='Berlin'
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 Beispiel: SELECT SUM(preis)
MIN (Feldname Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld Beispiel: SELECT MIN(preis)
MAX (Feldname Formel)	Ermittelt den größten aller Werte im angegebenen Feld Beispiel: SELECT MAX(preis)

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

d) Deutschland ist in zehn Postleitzahlregionen (0 bis 9, 1. Stelle der PLZ) unterteilt.

Korrekturrand

Erstellen Sie eine SQL-Abfrage für eine Liste, die für jede Postleitzahlregion (0 bis 9) den Gesamtumsatz ausweist. Die Liste soll nach Gesamtumsatz absteigend sortiert werden. 7 Punkte

7 Punkte

Beipielausgabe

Gebiet	Umsatz
5	154.000
8	147.000
4	132.000
...	

Hinweis: Die Kopfzeile wird nicht mit ausgegeben.

[illegible]

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.

7

