

1

Ganzheitliche Aufgabe I Fachqualifikationen

Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.). Wird eine bestimmte Anzahl verlangt (z. B. „Nennen Sie fünf Merkmale ...“), so ist bei Aufzählung von fünf richtigen Merkmalen die volle vorgesehene Punktzahl zu geben, auch wenn im Lösungshinweis mehr als fünf Merkmale genannt sind. Bei Angabe von Teilpunkten in den Lösungshinweisen sind diese auch für richtig erbrachte Teilleistungen zu geben.

In den Fällen, in denen vom Prüfungsteilnehmer

- keiner der fünf Handlungsschritte ausdrücklich als „nicht bearbeitet“ gekennzeichnet wurde,
- der 5. Handlungsschritt bearbeitet wurde,
- einer der Handlungsschritte 1 bis 4 deutlich erkennbar nicht bearbeitet wurde,

ist der tatsächlich nicht bearbeitete Handlungsschritt von der Bewertung auszuschließen.

Ein weiterer Punktabzug für den bearbeiteten 5. Handlungsschritt soll in diesen Fällen allein wegen des Verstoßes gegen die Formvorschrift nicht erfolgen!

Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1 =	100 – 92 Punkte	Note 2 =	unter	92 – 81 Punkte
Note 3 =	unter 81 – 67 Punkte	Note 4 =	unter	67 – 50 Punkte
Note 5 =	unter 50 – 30 Punkte	Note 6 =	unter	30 – 0 Punkte

1. Handlungsschritt (25 Punkte)

a) 4 Punkte

- Daten und Methoden innerhalb einer Struktur zusammenfassen
- Kontrollierter Zugriff auf Daten und Methoden über Schnittstelle (Geheimnisprinzip)

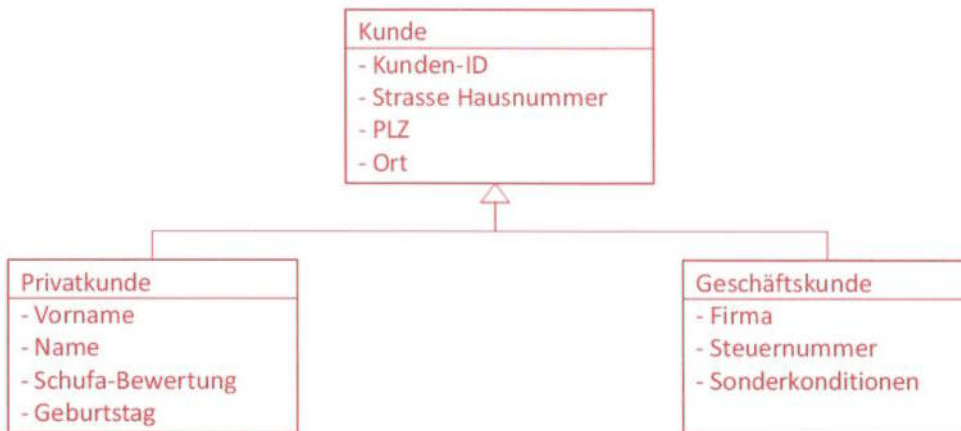
ba) 4 Punkte

- Objekte aus Einzelteilen, die auch unabhängig vom zusammenfassenden Objekt bestehen können
- Die Freigabe des Vertrag-Objektes bewirkt nicht die Freigabe der Fahrzeug- und Kundenobjekte.

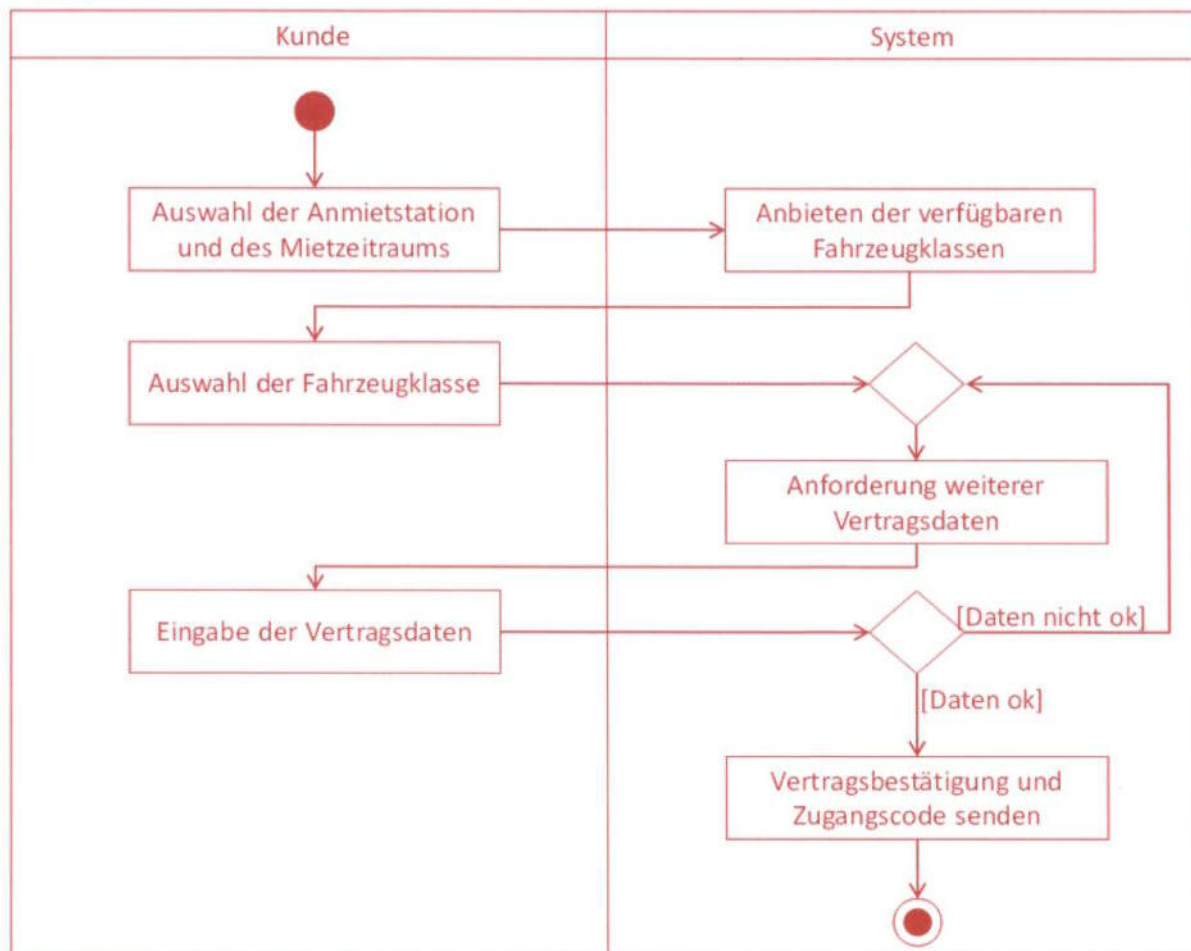
bb) 4 Punkte

- Die Existenz der Teile sind vom Ganzen abhängig, d. h. ohne das Ganze können die Teile nicht existieren.
- Die Freigabe des Vertrag-Objektes bewirkt die Freigabe des Rechnung-Objektes.

c) 3 Punkte



d) 10 Punkte

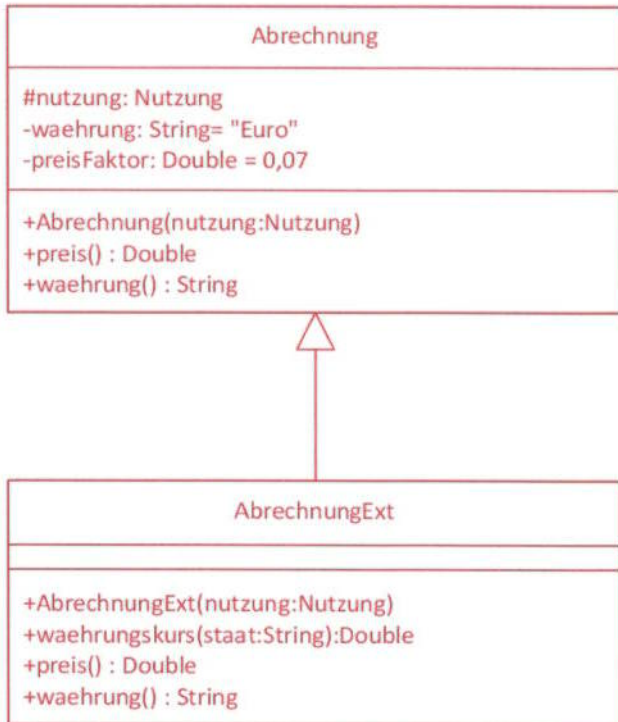


2. Handlungsschritt (25 Punkte)

a) 5 Punkte

Polymorphie bedeutet, dass bei einer Referenz auf ein Objekt vom Typ einer Basisklasse oder Interface erst zur Laufzeit entschieden wird, welche Methode einer abgeleiteten oder implementierenden Klasse aufgerufen wird (Dynamisches Binden).

b) 5 Punkte



ca) 3 Punkte

```
c#
public AbrechnungExt(Nutzung nutzung)
    : base(nutzung)
{ }
```

Beispiel für Pseudocode

```
öffentlich AbrechnungExt(nutzung: Nutzung)
    Basisklassenkonstruktor(nutzung : Nutzung)
```

Ende AbrechnungExt

cb) 6 Punkte

```
c#
public override double Preis()
{
    return base.Preis() * Waehrungskurs(nutzung.Kunde.Staat);
}
```

Beispiel für Pseudocode

```
öffentlich preis() : double
    Rückgabe preis() aus Basisklasse * waehrungskurs(nutzung.kunde.staat)
```

Ende preis

cc) 6 Punkte

```
c#
public override string Waehrung()
{
    switch (nutzung.Kunde.Staat)
    {
        case "Schweden":
            return "SEK";
        case "Norwegen":
            return "NOK";
        case "Dänemark":
            return "DKK";
        case "Island":
            return "ISK";
        default:
            return base.Waehrung();
    }
}
```

Beispiel für Pseudocode

```
öffentlich waehrung() : String
    wenn nutzung.kunde.staat == "Schweden"
        Rückgabe "SEK"
    Ende wenn
    wenn nutzung.kunde.staat == "Norwegen"
        Rückgabe "NOK"
    Ende wenn
    wenn nutzung.kunde.staat == "Dänemark"
        Rückgabe "DKK"
    Ende wenn
    wenn nutzung.kunde.staat == "Island"
        Rückgabe "ISK"
    Ende wenn
    Rückgabe waehrung() aus Basisklasse
Ende waehrung
```

3. Handlungsschritt (25 Punkte)

Pseudocode

```
boolean pruefeID(string id)

int sum := 0
int sum2 := 0
int anzahlZiffern := Laenge(id)
int ziffer
int i
char stelle

Für i von 0 bis anzahlZiffern - 2
    stelle := Zeichen an der Stelle i von id
    ziffer := wandle stelle in Zahl

    Wenn i modulo 2 = 0 dann
        ziffer := ziffer * 2

        Wenn ziffer > 9 dann
            ziffer := ziffer - 9
        Ende Wenn

    Ende Wenn

    sum := sum + ziffer

Ende Für

sum2 := sum

Wenn sum modulo 10 <> 0 dann
    sum2 := sum + 10 - sum modulo 10
Ende Wenn

stelle := Zeichen an der Stelle anzahlZiffer - 1 von id
ziffer := Wandle stelle in Zahl

Rueckgabe sum2 - sum = ziffer

Ende pruefeID
```

pruefelD(id : String) : boolean

Übergabe der id als String

sum:=0, sum2:=0, anzahlZiffern:=Länge der id

von i:=0 solange i<=anzahlZiffern-2, Schritt 1

stelle:=Zeichen an der Stelle i

ziffer:=stelle als Integerwert

i modulo 2 = 0

ja

nein

ziffer:=ziffer*2

ziffer > 9

ja

nein

Ø

ziffer:=ziffer-9

Ø

sum:=sum+ziffer

sum2:=sum

sum modulo 10 <> 0

ja

nein

sum2:=sum+10-(sum modulo 10)

Ø

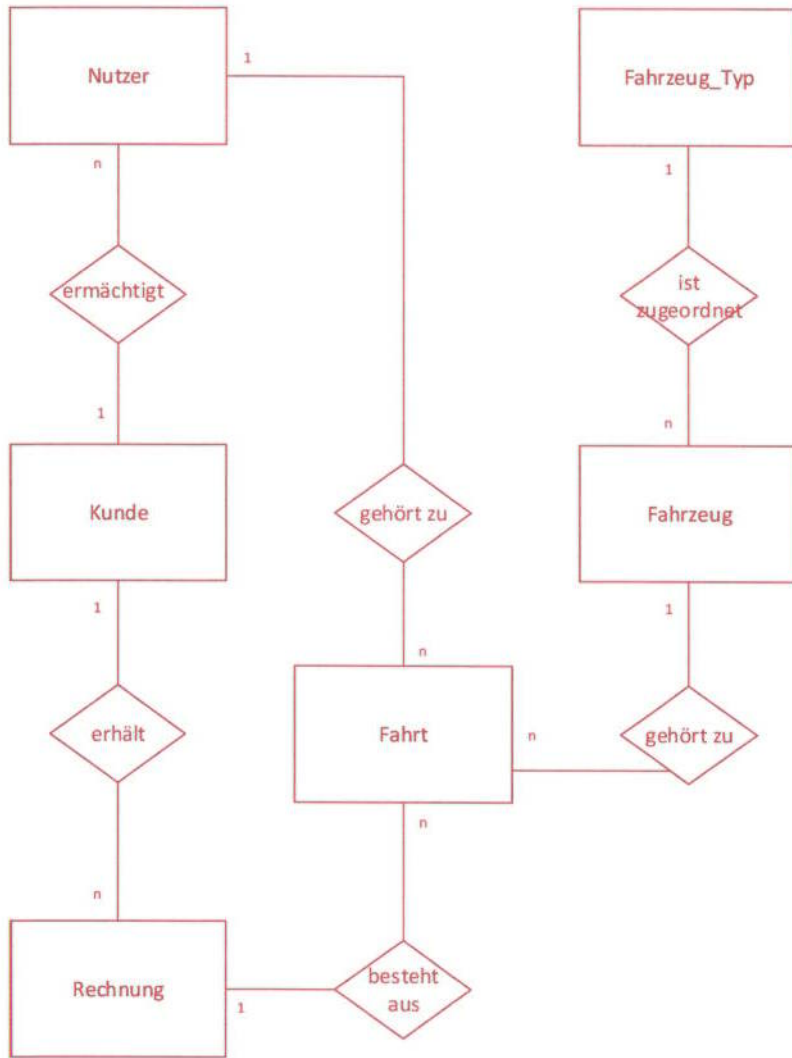
stelle:=Zeichen an der Stelle id-Länge-1

ziffer:=stelle als Integerwert

Rückgabe sum2-sum=ziffer

4. Handlungsschritt (25 Punkte)

a) 17 Punkte



b) 8 Punkte

Fahrzeug_Typ
FahrzeugTyp_ID (PK)

Kunde
Kunden_ID (PK)

Rechnung
Rechnungsnummer_ID (PK)
Kunden_ID (FK)

Fahrzeug
Fahrzeug_ID (PK)
FahrzeugTyp_ID (FK)

Nutzer
Nutzer_ID (PK)
Kunden_ID (FK)

Fahrten
Fahrten_ID (PK)
Nutzer_ID (FK)
Fahrzeug_ID (FK)
Rechnungsnummer_ID (FK)

5. Handlungsschritt (25 Punkte)

a) 4 Punkte

UPDATE Artikel

SET Art_Markierung = (CASE WHEN Art_MwStSatz = 7 THEN 1 ELSE 0 END);

b) 6 Punkte

SELECT

(SELECT COUNT(Rg.Rg_ID) FROM Rechnung AS Rg
WHERE Rg.Rg_KdId = Kd_ID AND YEAR(Rg.Rg_Datum) = 2015) AS AnzRechnung,
Kd_Id, Kd_Name, Kd_Vorname, Kd_Strasse, Kd_PLZ, Kd_Ort, Kd_ZbID

FROM Kunde AS Kd

INNER JOIN Zahlungsbedingung AS Zb

ON Zb.Zb_Id = Kd.Kd_ZbID AND Zb.Zb_SkontoProzent > 3;

c) 4 Punkte

SELECT *

FROM Kunde

WHERE Kd_Name LIKE '%oe%' OR Kd_Name LIKE '%ö%' OR Kd_Name LIKE '%õe%';

d) 4 Punkte

SELECT COUNT(Kd_Id) AS AnzKunden, Kd_Ort AS Ort

FROM Kunde

GROUP BY Kd_Ort ORDER BY AnzKunden DESC;

e) 7 Punkte

SELECT 'Kunde' AS Tabellenname

, [Kd_Id] AS ID

, [Kd_Name] AS Name

, [Kd_PLZ] AS PLZ

, [Kd_Ort] AS Ort

FROM Kunde

UNION ALL

SELECT 'Lieferant'

, [Lf_Id]

, [Lf_Firma]

, [Lf_PLZ]

, [Lf_Ort]

FROM Lieferant

ORDER BY Name DESC;