

# 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.

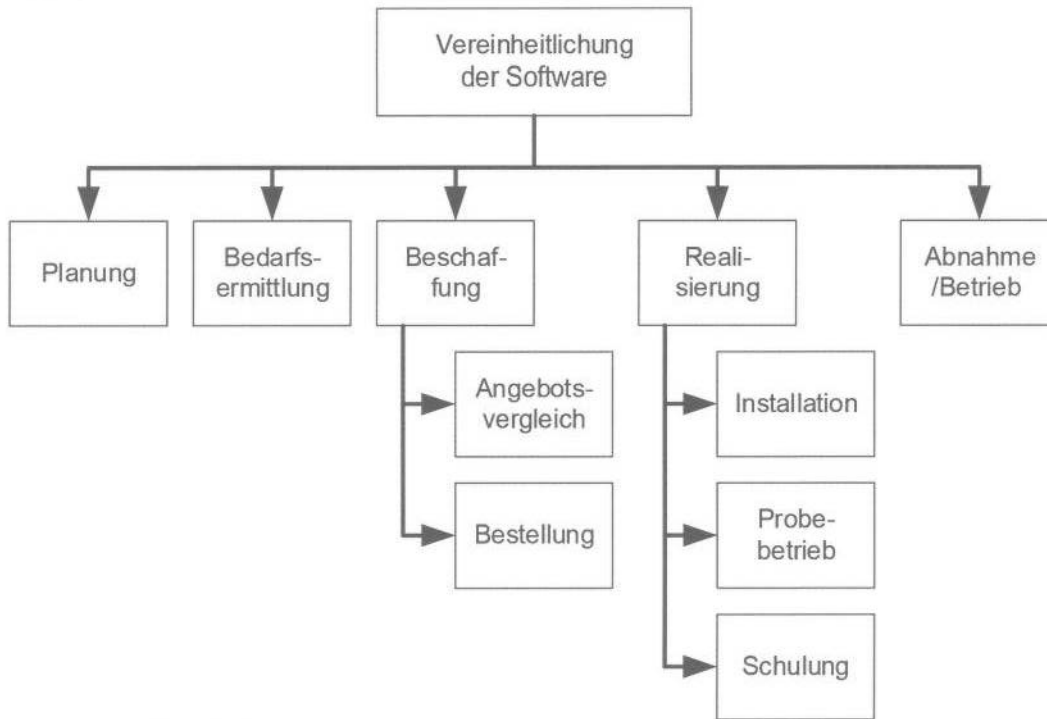
Ein weiterer Punktabzug für den bearbeiteten 5. Handlungsschritt soll in diesen Fällen allein wegen des Verstoßes gegen die Formvorschrift nicht erfolgen!

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

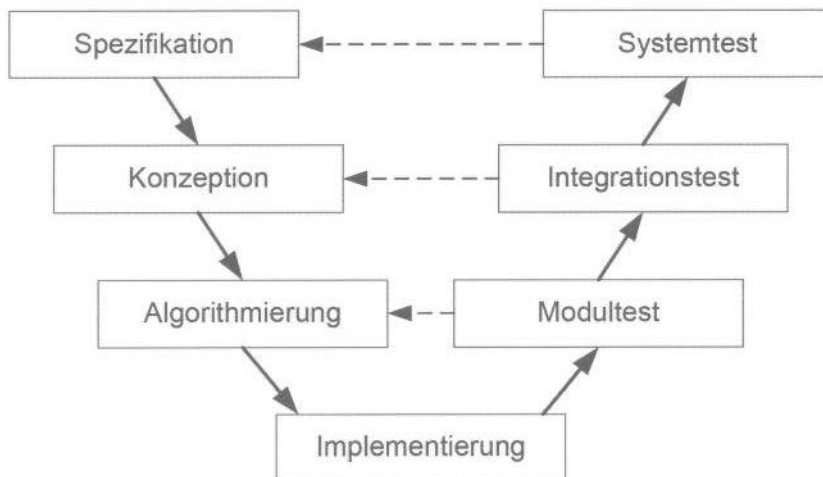
# 1. Handlungsschritt (25 Punkte)

a) 5 Punkte



Andere Lösungen sind möglich.

ba) 7 Punkte, 7 x 1 Punkt je Phase



bb) 3 Punkte, 6 x 0,5 Punkte je Nennung

Bezeichnung	Beschreibung	Dokumentation
Algorithmierung	Module algorithmieren	Beispiel: Struktogramm
Implementierung	Module implementieren	Quellcode
Integrationstest	Zusammenwirken der Module testen	Testbericht
Konzeption	System konzipieren	Pflichtenheft
Modultest	Module testen	Testbericht/-protokoll
Spezifikation	Systemanforderung spezifizieren	Lastenheft
Übergabe	System übergeben	Übergabeprotokoll

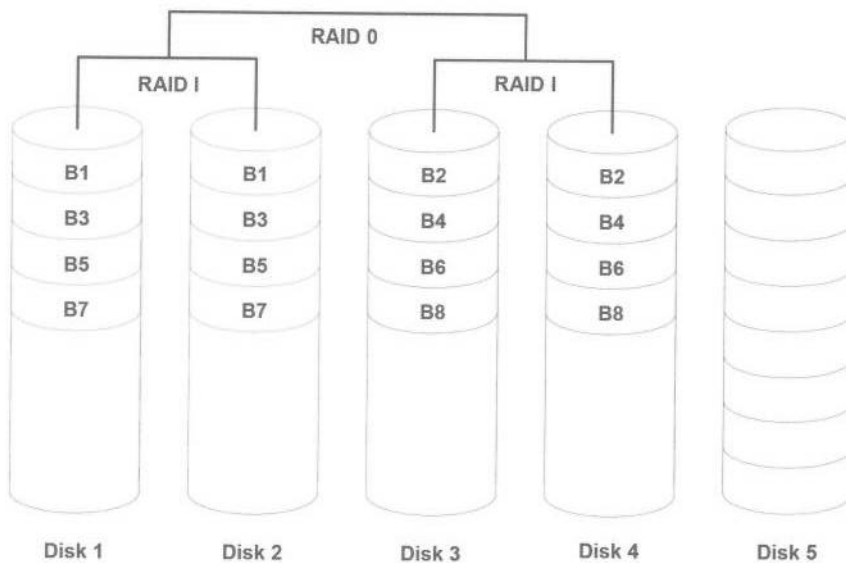
ca) 5 Punkte

- 3 TB Nettokapazität ( $4 \times 1,5 / 2$ )
- RAID 0: keine Redundanz.
- RAID 1: volle Redundanz.

cb) 5 Punkte

4 Punkte, 2 x 2 Punkte je RAID I

1 Punkt, RAID 0



## 2. Handlungsschritt (25 Punkte)

erstelleLadungsliste(ladungsliste : zweidimensionales Array vom Typ Integer, nutzlast\_kg : Double, maxAnzahl\_Palette : Integer)

anzahlLKW := 10

lkwNr := 1

beladen := false

**solange** beladen = false **und** lkwNr <= anzahlLKW

beladen := true

zuladung\_kg := 0

zuladung\_Palette := 0

**für** i = 0 **bis** i = Anzahl Zeilen von ladungsliste - 1

**wenn** zuladung\_kg + ladungsliste[i][0] <= nutzlast\_kg  
**und** zuladung\_Palette < maxAnzahl\_Palette  
**und** ladungsliste[i][1] = 0

zuladung\_kg := zuladung\_kg + ladungsliste[i][0]

zuladung\_Palette := zuladung\_Palette + 1

ladungsliste[i][1] := lkwNr

beladen := false

**ende wenn**

**ende für**

lkwNr := lkwNr + 1

**ende solange**

### 3. Handlungsschritt (25 Punkte)

```
public Angebot ErstelleAngebot(string kundennummer, string startOrt,
    string zielOrt, double gewicht, DateTime versandTermin)
{
    Angebot anbot = new Angebot();
    Route[] r =
        GetRouten(startOrt, zielOrt, versandTermin);
    double minRoutenPreis = 0.00;
    int indexGuenstigsteRoute = 0;

    for (int i = 0; i < r.Length; i++)
    {
        double routenPreis = 0.00;
        for (int j = 0; j < r[i].GetStrecken.Length; j++)
        {
            routenPreis += r[i].GetStrecken[j].Preis;
        }
        if (minRoutenPreis == 0.00)
        {
            minRoutenPreis = routenPreis;
            indexGuenstigsteRoute = i;
        }
        else
        {
            if (minRoutenPreis > routenPreis)
            {
                minRoutenPreis = routenPreis;
                indexGuenstigsteRoute = i;
            }
        }
    }
    anbot.Angebotsnummer = ErzeugeAngebotsnummer();
    anbot.Kundennummer = kundennummer;

    anbot.PreisOhneRabatt =
        GetPreisOhneRabatt(minRoutenPreis, gewicht);

    anbot.RabattProzentsatz =
        Math.Max( GetKundenRabattInProzent(kundennummer),
            GetGewichtsRabattInProzent(gewicht) );

    anbot.PreisMitRabatt =
        anbot.PreisOhneRabatt -
        (anbot.PreisOhneRabatt * anbot.RabattProzentsatz / 100);

    anbot.Route = r[indexGuenstigsteRoute];

    return anbot;
}
```

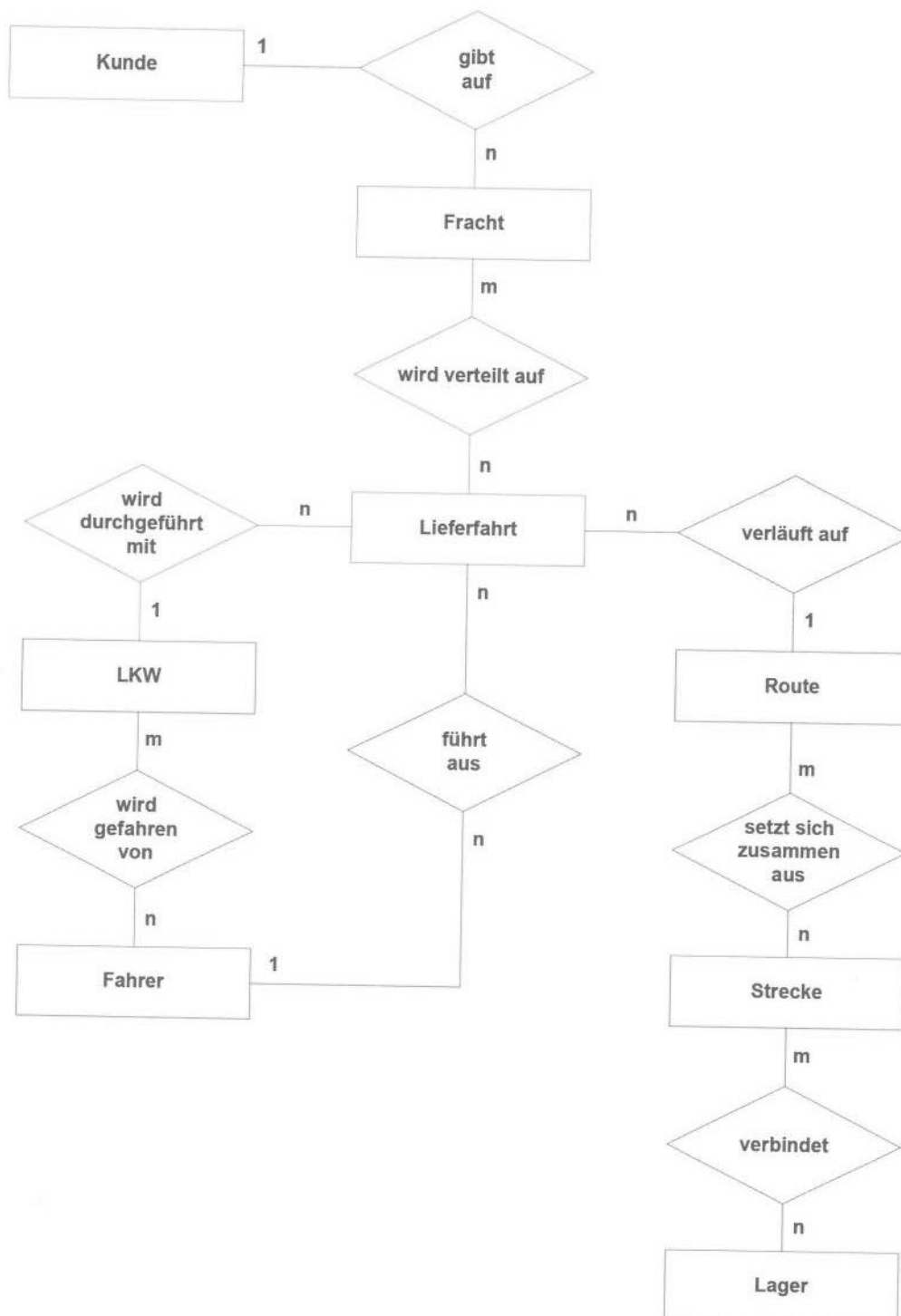
#### 4. Handlungsschritt (25 Punkte)

a) 23 Punkte

7 Punkte, 7 x 1 Punkt je Entitätstyp

8 Punkte, 8 x 1 Punkt je Beziehung

8 Punkte, 8 x 1 Punkt je Kardinalität



b) 2 Punkte

n : m-Beziehungen müssen aufgelöst werden. Es sind Zwischentabellen notwendig.

Bei 1 : n Beziehungen muss der Primärschlüssel der Tabelle der 1-Seite als Fremdschlüssel der Tabelle der n-Seite aufgeführt sein.

## 5. Handlungsschritt (25 Punkte)

a) 4 Punkte

```
SELECT F.F_ID, F.F_Fa_ID, F.F_Au_ID, Fa.Fa_Name  
FROM Fahrten AS F LEFT JOIN Fahrer AS Fa ON F.F_Fa_Id = Fa.Fa_Id;
```

b) 6 Punkte

```
DELETE FROM KUNDE  
WHERE (SELECT COUNT(Au_Id) FROM Auftrag WHERE Au_Kd_Id = Kd_Id ) = 0;
```

c) 5 Punkte

```
SELECT Ts.Ts_ID, Ts.Ts_Strecke, Fa.Fa_Name, Au.Au_AuftragNr, Kd.Kd_ID,  
Kd.Kd_Firma, Kd.Kd_Strasse, Kd.Kd_PLZ, Kd.Kd_Ort  
FROM ( ( ( Teilstrecke AS Ts  
    INNER JOIN Fahrten AS F ON Ts.Ts_F_Id = F.F_Id )  
    INNER JOIN Fahrer AS Fa ON F.F_Fa_Id = Fa.Fa_Id )  
    INNER JOIN Auftrag AS Au ON F.F_Au_Id = Au.Au_Id )  
    INNER JOIN Kunde AS Kd ON Au.Au_Kd_Id = Kd.Kd_Id;
```

d) 10 Punkte

```
SELECT Kd.Kd_ID, Kd.Kd_Firma, Kd.Kd_Strasse, Kd.Kd_PLZ, Kd.Kd_Ort,  
( SELECT COUNT(Au.Au_Id)  
  FROM Auftrag AS Au  
    WHERE Au.Au_Kd_Id = Kd.Kd_Id ) AS AnzahlAuftrag,  
( SELECT COUNT(F.F_Id)  
  FROM Fahrten AS F, Auftrag AS Au  
    WHERE Au.Au_Id = F.F_Au_Id AND Au.Au_Kd_Id = Kd.Kd_Id ) AS AnzahlFahrten,  
( SELECT Sum(Ts.Ts_Strecke)  
  FROM Teilstrecke AS Ts, Auftrag AS Au, Fahrten AS F  
    WHERE Au.Au_Id = F.F_Au_Id AND Au.Au_Kd_Id = Kd.Kd_Id AND Ts.Ts_F_Id = F.F_Id ) AS  
SumStrecke  
FROM Kunde AS Kd  
ORDER BY Kd.Kd_PLZ DESC;
```

Andere Lösungen sind möglich.