

# 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) 15 Punkte

Eine alternative Lösung ist auch als richtig zu bewerten.

Funktion generateCode(kdnr : Integer, datum : Date) : string

kwJahr : String

kw : Integer

jahr : Integer

herkunft : String

a1, a2, a3 : Integer

nummer : Integer

kwJahr = getKWJahr(datum)

kw = (Integer)(Stelle 1 bis 2 von kwJahr)

jahr = (Integer)(Stelle 3 bis 6 von kwJahr)

herkunft = getHerkunft(kdnr)

a1 = (Integer)(1. Zeichen von herkunft)

a2 = (Integer)(2. Zeichen von herkunft)

a3 = (Integer)(3. Zeichen von herkunft)

nummer = ( ( (a1 \* 91 + a2) \* 91 + a3 ) \* 54 + kw ) \* 2300 + jahr

Rückgabe code

b) 10 Punkte

Eine alternative Lösung ist auch als richtig zu bewerten.

Funktion getLand(code : Integer) : String

kw, jahr : Integer

c1, c2, c3 : Character

jahr = code modulo 2300

code = code - jahr

code = code / 2300

kw = code modulo 54

code = code - woche

code = code / 54

c3 = (Character)(code modulo 91)

code = code - code modulo 91

code = code / 91

c2 = (Character)(code modulo 91)

code = code - code modulo 91

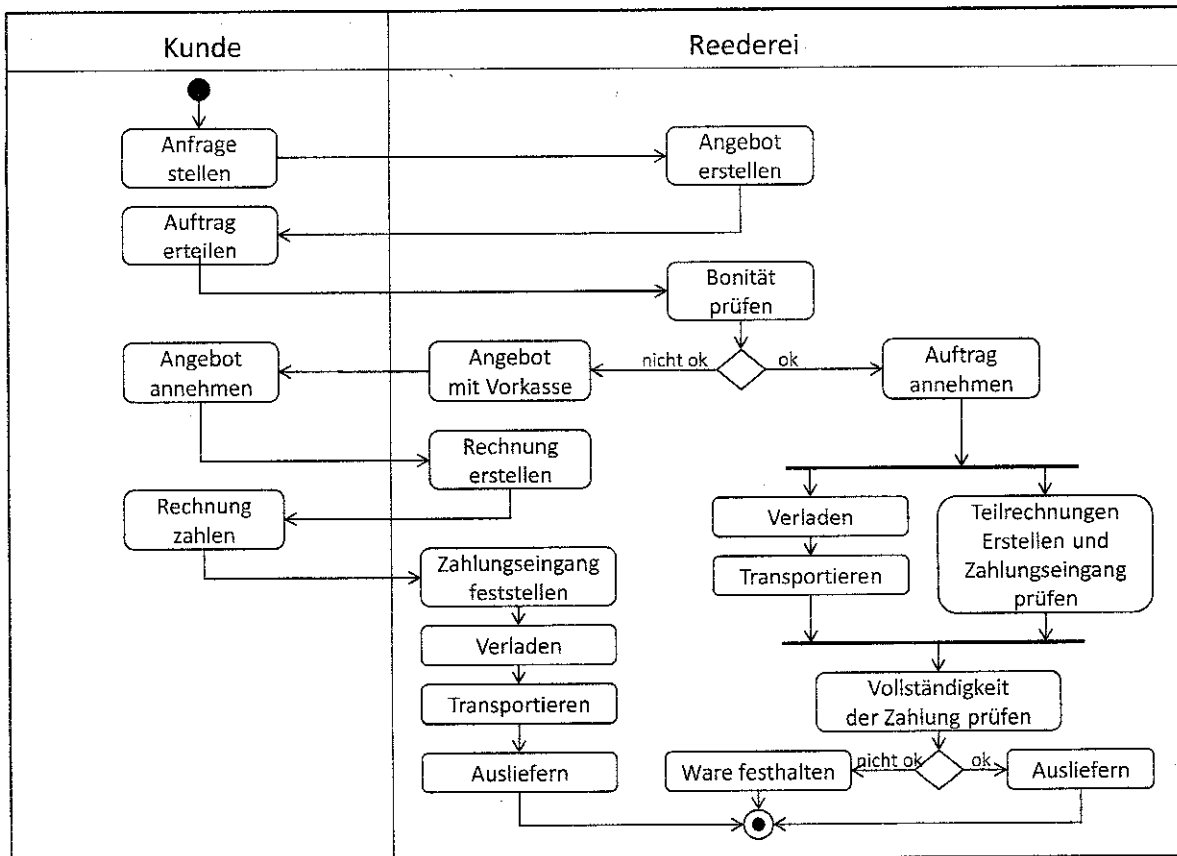
code = code / 91

c1 = (Character)(code)

Rückgabe c1 + c2 + c3

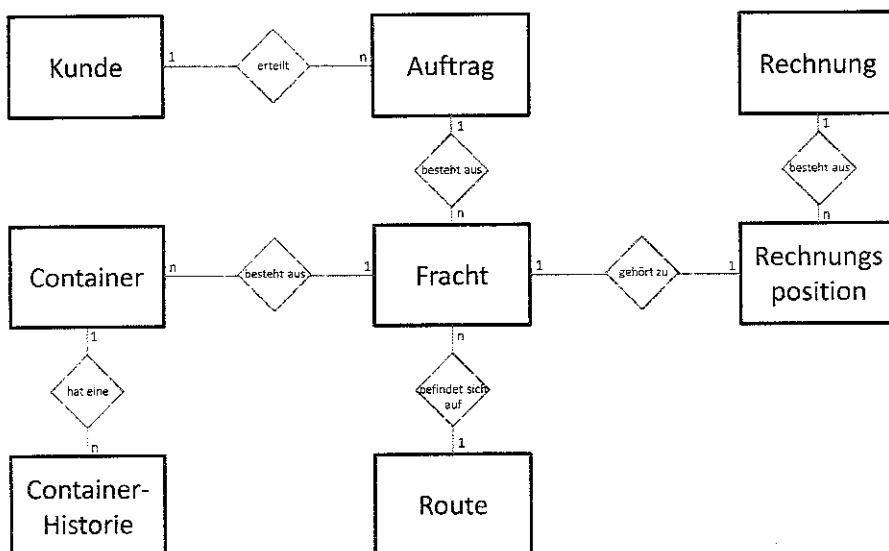
## 2. Handlungsschritt (25 Punkte)

Eine alternative Lösung ist auch als richtig zu bewerten.



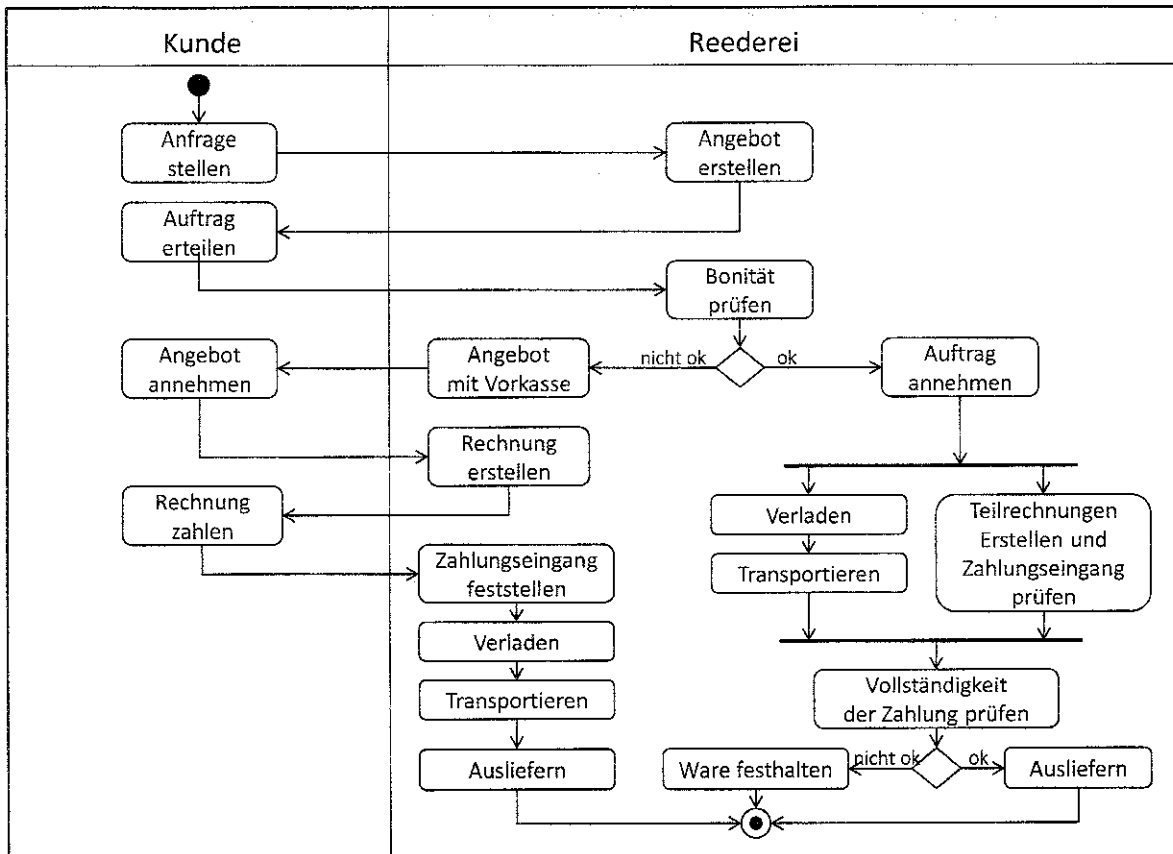
## 3. Handlungsschritt (25 Punkte)

Eine alternative Lösung ist auch als richtig zu bewerten.



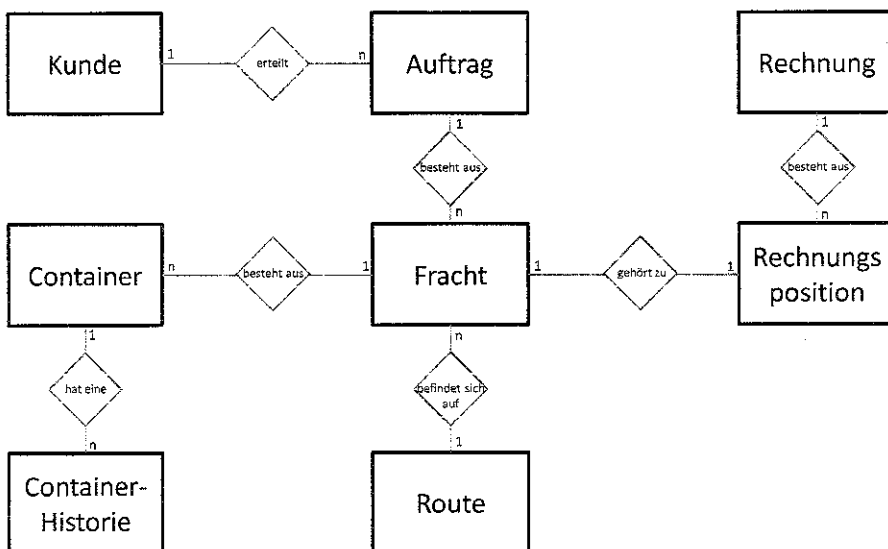
## 2. Handlungsschritt (25 Punkte)

Eine alternative Lösung ist auch als richtig zu bewerten.



## 3. Handlungsschritt (25 Punkte)

Eine alternative Lösung ist auch als richtig zu bewerten.



#### 4. Handlungsschritt (25 Punkte)

Eine alternative Lösung ist auch als richtig zu bewerten.

```
erstelleStauplan(Container[] : containerListe)

// Deklarationen
anzahlBay[ ] : array [3] of integer           // Anzahl Container in einer Bay
gewichtBay[ ] : array [3] of integer          // Beladungsgewicht einer Bay
bayTier[ ] [ ] : array [3] [4] of Container   // Tiers in Bay
pos, gewichtBay_pos : integer                // Hilfsvariablen
i, j : integer                               // Zähler

// Vom ersten bis zum letzten Container in der Containerliste
Für i = 0 bis containerListe.laenge - 1
    // Prüfe, ob im Schiff noch ein freier Platz vorhanden ist
    pos := - 1
    Für j := 0 bis anzahlBay.laenge - 1
        Wenn anzahlBay[j] < 4 dann
            pos := j
            breche Schleife ab
        Ende wenn
    Ende für
    // Abbruch der Beladung, wenn Schiff voll beladen ist
    Wenn pos = -1 dann breche Schleife ab

    // Ermittle im Schiff den Laderaum mit der geringsten Gewichtsbeladung und der nicht voll belegt ist
    gewichtBay_pos := gewichtBay(pos)
    Für j = 0 bis anzahlBay.laenge - 1
        Wenn gewichtBay[j] < gewichtBay_pos und anzahlBay[j] < 4 dann
            pos := j
            gewichtBay_pos = gewichtBay[pos]
        Ende wenn
    Ende Für

    // - Weise den Container der ausgewählten Bay und der freien Tier zu:
    // - Erhöhe das Beladungsgewicht der Bay um das Gewicht des Containers
    // - Erhöhe die Anzahl Container in der Bay
    bayTier[pos][anzahlBay[pos]] := containerListe[i]
    gewichtBay[pos] := gewichtBay[pos] + containerListe[i].gewicht
    anzahlBay[pos] := anzahlBay[pos] + 1

Ende für // Containerliste

// Stauplan ausgeben
Für i := 0 bis bayTier.Länge - 1           // Gehe alle Bays durch
    drucke „Bay „ + (i + 1)
    Für j := 0 to bayTier[j].länge - 1 // Gehe alle Tiers einer Bay durch
        Wenn bayTier[i][j] = null dann
            drucke „ Tier “ + (j + 1)
            + „ CNr “ + bayTier[i][j].nr +
            + „ “ + bayTier[i][j].gewicht + „ t“
        Ende Wenn
    Ende für // Tiers
    drucke „ Beladung: “ + gewichtBay[i] + „ t“
Ende für // Bays
```