



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 5 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) 2 Punkte, 2 x 1 Punkt

- Motivation
- Information
- Kennenlernen
- Darstellung der Ausgangssituation
- Darstellung des angestrebten Ziels
- u. a.

b) 4 Punkte, 2 x 2 Punkt

- Keine Reisekosten
- Kürzere Einarbeitungszeit, da Mitarbeiter/-innen der Krake AG die Abläufe kennen
- Know-how bleibt in der Unternehmung

c) 2 Punkte, 2 x 1 Punkt

- Genaue Anpassung an Abläufe der Krake AG möglich
- Unabhängigkeit von externem Hersteller
- Keine Wartungsgebühren

da) 5 Punkte

- Ist-Analyse
- Sollkonzept
- Implementierung
- Tests
- Schulung

db) 5 Punkte

- Lastenheft
- Pflichtenheft
- Dokumentation
- Benutzerhandbuch
- Testprotokolle
- Abnahmeprotokoll

ea) 4 Punkte 2 x 2

Whiteboxtest:

- Auffinden von Logikfehlern
- Optimierung eines Programmablaufs
- Feststellung von Überdeckungsgraden (z. B. Anweisungsüberdeckung)

Blackboxtest:

- um die Funktionalität zu prüfen

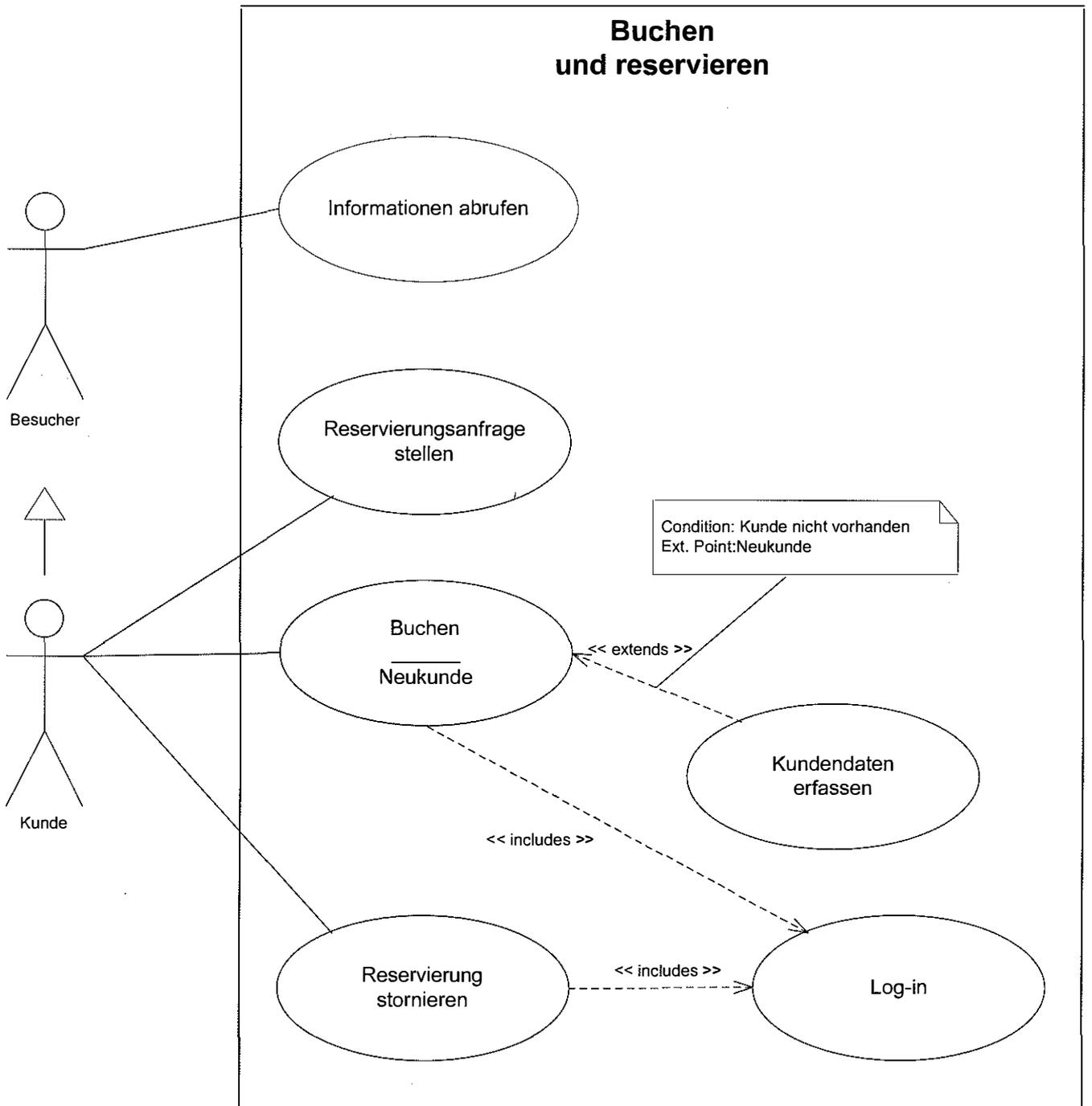
eb) 3 Punkte

- Frühes Testen möglich
- Nachvollziehbarkeit
- Schnelle Durchführung von Tests nach Änderungen der Software, z. B. Legal Changes

Alternative, in sich schlüssige Lösungen sind auch als richtig zu bewerten.

2. Handlungsschritt (25 Punkte)

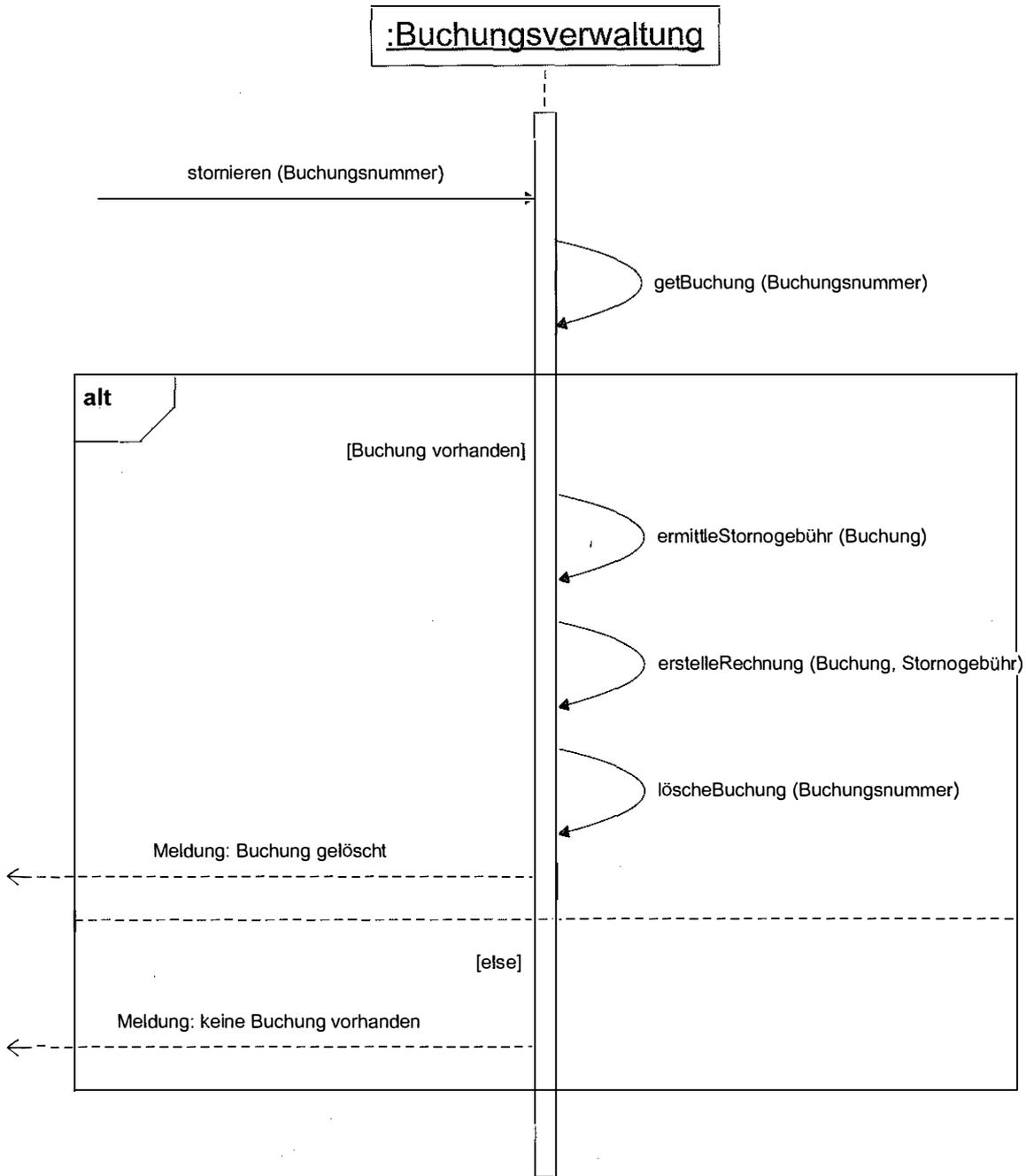
a) 15 Punkte



Fortsetzung 2. Handlungsschritt →

Fortsetzung 2. Handlungsschritt

b) 10 Punkte



Andere Lösungen sind möglich.

3. Handlungsschritt (25 Punkte)

Kunde
Kd_Id (PK)
Kd_Nr
Kd_Name
Kd_Vorname
Kd_Adr_Id (FK)

Reservierung
Res_Id (PK)
Res_Kd_Id (FK)
Res_Nr
Res_Datum
Res_LieferAdr_Id (FK)

ResPosition
ResPos_Id (PK)
ResPos_Res_Id (FK)
ResPos_PosNr
ResPos_PreisTag
ResPos_LieferPreis
ResPos_VonDatum
ResPos_BisDatum
ResPos_G_ID (FK)

Geraet
G_Id (PK)
G_GTyp_Id (FK)
G_Bez
G_Nr

GeraeteTyp
GTyp_Id (PK)
GTyp_Bez
GTyp_PreisTag
GTyp_Lieferpreis

Adresse
Adr_Id (PK)
Adr_Strasse
Adr_PLZ
Adr_Ort

Andere Lösungen sind möglich.

4. Handlungsschritt (25 Punkte)

a) 2 Punkte

```
UPDATE Mitarbeiter SET Tagesarbeitszeit = 7
WHERE MA_Id = 812;
```

b) 2 Punkte

```
SELECT COUNT(Ma_Id) FROM Mitarbeiter
WHERE Tagesarbeitszeit < 7.33;
```

c) 4 Punkte

```
SELECT Mitarbeiter.Ma_Id, Nachname, Fehlgrund, SUM(Fehltage)
FROM Fehlzeit, Mitarbeiter
WHERE Fehlzeit.Ma_Id = Mitarbeiter.Ma_Id
GROUP BY Mitarbeiter.Ma_Id, Nachname, Fehlgrund;
```

d) 6 Punkte

```
SELECT Mitarbeiter.Ma_Id, Nachname, Vorname, SUM(EinsatzBis_Zeit -- EinsatzVon_Zeit)
FROM Mitarbeiter LEFT JOIN Einsatz ON Mitarbeiter.Ma_Id = Einsatz.Ma_Id
WHERE MONTH(Datum) = 2 AND YEAR(Datum) = 2009;
```

e) 6 Punkte

```
SELECT Mitarbeiter.Ma_Id, Nachname, Vorname, UrlaubsanspruchJahr- (SELECT
SUM(Fehltage) FROM Fehlzeit WHERE Fehlzeit.Ma_Id = Mitarbeiter.Ma_Id AND Fehlgrund
= 'Urlaub' AND YEAR(Von_Datum) = 2009)
FROM Mitarbeiter;
```

f) 5 Punkte

```
SELECT * FROM Mitarbeiter
WHERE Nachname LIKE 'SCH%' AND YEAR(NOW)-YEAR(Geb_Datum) >30 ORDER BY Geb_Datum DESC, Nachname;
```

Andere Lösungen sind möglich.

5. Handlungsschritt (25 Punkte)

verfügTest(Gerätetyp, Entleihbeginn, Entleihdauer)

```
geräteID[] bestandsListe = getGeräteListe(Gerätetyp) //Einlesen der GeräteIDs eines Gerätetyps

found = false
zähler = 0

solange1 (found == false) und zähler < (Anzahl der Einträge von bestandsliste))
    buchungsdat[] buchungsListe=getResDat(bestandsliste[zähler])
    // Wenn keine Reservierung vorhanden
    wenn1 (Länge von buchungsListe == 0)
        found = true
    ende wenn1
    // Wenn nur eine Reservierung vorhanden
    // Abfrage, ob Mietzeitraum vor Beginn der ersten Reservierung endet oder nach deren Ende beginnt
    wenn2 (Länge von buchungsListe == 1)
        wenn3 (buchungsListe[0].vonDat > (Entleihbeginn + Entleihdauer)
            oder buchungsListe [0].bisDat < Entleihbeginn)
            found = true
        ende wenn3
    ende wenn2
    // Wenn mehrere Reservierungen vorhanden
    wenn4 (Länge von buchungsListe > 1)
        // Abfrage, ob Mietzeitraum vor Beginn der ersten Reservierung endet
        wenn5 (buchungsListe[0].vonDat > (Entleihbeginn + Entleihdauer)
            found = true
        ende wenn5
        // Abfrage, ob Mietzeitraum zwischen zwei Reservierungen passt
        i = 0
        solange2 (i < Länge von buchungsListe - 1 und found == false)
            wenn6 (buchungsListe[i].bisDat < Entleihbeginn
                und buchungsListe[i+1].vonDat > Entleihbeginn + Entleihdauer)
                found = true
            ende wenn6
            i = i + 1
        ende solange2
        // Abfrage, ob gewünschter Mietzeitraum nach letzter Reservierung beginnt
        wenn7 (found == false und buchungsListe[i].bisDat < Entleihbeginn)
            found = true
        ende wenn7
    ende wenn4

    wenn8 (found == false)
        zähler = zähler + 1
    ende wenn8
ende solange1

wenn (found == false)
    Rückgabe 0
sonst
    Rückgabe bestandsListe[zähler]
ende wenn
```

ende funktion

Andere Lösungen sind möglich.