Abschlussprüfung Winter 2022/23
Lösungshinweise

Fachinformatiker/Fachinformatikerin Anwendungsentwicklung (AO 2020) 1201

2 Entwicklung und Umsetzung von Algorithmen

Teil 2 der Abschlussprüfung

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.).

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. Aufgabe (25 Punkte)

a) 10 Punkte

```
void sort(raeume: Array von Raum)
      für i = 0, i < raeume.length-1; i++
            für j = 0; j < raeume.length – i -1; j++
                  wenn raeume[j].getBelegung() >
                              raeume[j+1].getBelegung()
                        Raum r = raeume[j]
                        raeume[j] = raeume[j+1]
                        raeume[j+1] = r
                  ende wenn
            ende für
      ende für
ende sort
```

b) 5 Punkte

```
vergleicheMitarbeiter(Mitarbeiter m1, Mitarbeiter m2): Integer
      diff: Double
      diff =m1.getGehalt() – m2.getGehalt()
      wenn diff == 0
            Rückgabe 0
      Ende wenn
      Wenn diff > 0
            Rückgabe 1
      Sonst
            Rückgabe -1
      ende wenn
ende vergleicheMitarbeiter
```

Hinweis: Eine Lösung mit Rückgabe m1.getGehalt() – m2.getGehalt() führt zu einem Typkonflikt zwischen Integer und Double. Eine Typkonvertierung von Gehalt zu Integer zu nicht richtigen Ergebnissen!

c) 3 Punkte

Eine generische Klasse ist eine typparametrisierte Klasse, das heißt Methoden und Variablen werden für einen vorgegebenen, aber beliebigen Typ definiert. Für den vorgegebenen Typ können Einschränkungen formuliert werden.

d) 7 Punkte

```
void sort(liste: List<T>, Function<T> f)
            für i = 0, i < liste.size()-1; i++
                  für j = 0; j < liste.size() – i -1; j++
                        wenn f(liste.get(j), liste.get(j+1)) > 0
                              T r = liste.get(j)
                              liste.set(j, liste.get(j+1))
                              liste.set(j+1, r)
                        ende wenn
                  ende für
            ende für
      ende sort
```

2. Aufgabe (25 Punkte)

aa) 4 Punkte

Testfall	Dezimalzahl	Erwartetes Ergebnis	Ergebnis
1	7	111	111
2	11	1011	1101

ab) 4 Punkte

Die Ausgabe „`Ausgabe(rest);`" erfolgt vor dem rekursiven Aufruf „`Umrechnung(dezimalzahl);`", beim Abstieg in die Rekursion. Daher ist das Ergebnis in der falschen Reihenfolge.
Korrektur: Ausgabe nach dem rekursiven Funktionsaufruf.

Korrekte Alternativen sind ebenfalls als richtig zu werten.

ac) 3 Punkte

Um Anweisungsüberdeckung zu erreichen, muss durch die Auswahl der Testfälle sichergestellt werden, dass jede Anweisung mindestens 1-mal ausgeführt wird.
Beim gegebenen Programm genügt ein Testfall.

b) 4 Punkte

z. B.:
Vorteil: Rekursive Lösungen sind zum passenden Problem einfacher oder eleganter zu schreiben.

Nachteil: Rekursive Lösungen benötigen mehr Speicher zur Laufzeit.

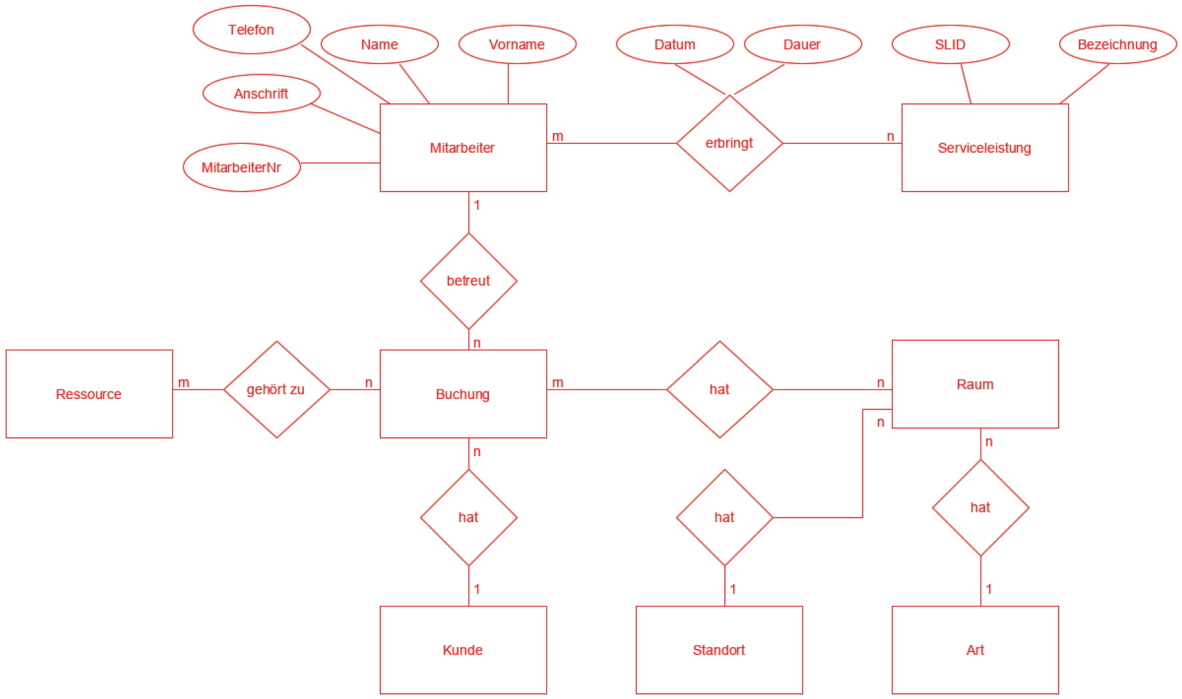
Korrekte Alternativen sind ebenfalls als richtig zu werten.

c) 10 Punkte

```
void Umrechnung(dezimalzahl: int)
BEGINN
      rest: int;
      i: int;
      List l();
      SOLANGE (dezimalzahl >0)
      BEGINN
            rest = dezimalzahl modulo 2;
            l.insert(rest);
            dezimalzahl = dezimalzahl div 2;
      ENDE
      ZÄHLE i VON l.size() BIS 0 SCHRITTWEITE -1
      BEGINN
            AUSGABE(l.element(i));
      ENDE
ENDE
```

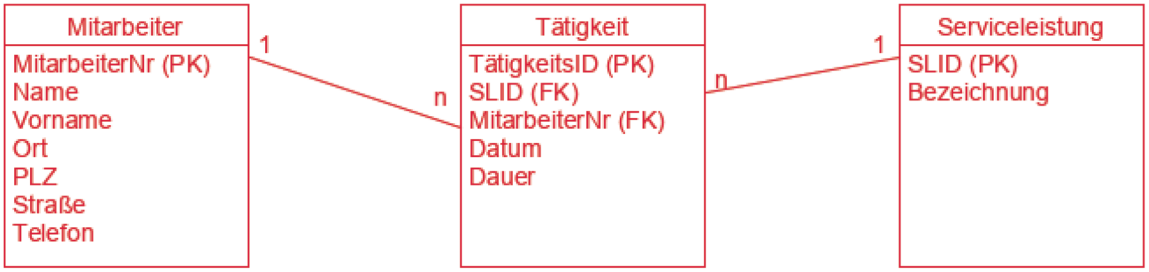
3. Aufgabe (25 Punkte)

a) 12 Punkte
1 Punkt für jeden richtigen Entitätstyp = 6 Punkte
1 Punkt für jede richtige Beziehung = 6 Punkte



Korrekte Alternativen sind ebenfalls als richtig zu werten.

b) 10 Punkte
Tabelle Mitarbeiter/Tätigkeit je 3 Punkte, Tabelle Serviceleistung 2 Punkte, je Beziehung 1 Punkt



Korrekte Alternativen sind ebenfalls als richtig zu werten.

c) 3 Punkte

Attribut	Datentyp
Dauer	Integer
PLZ	VarChar (String ,Text)
Bezeichnung	VarChar (String ,Text)

Korrekte Alternativen sind ebenfalls als richtig zu werten.

4. Aufgabe (25 Punkte)

a) 10 Punkte

```
SELECT Ma.Ma_ID AS ID
      ,Ma.Nachname
      ,Ma.Vorname
      ,Abt.Abteilung
      ,SUM(Fz.FehltageSum) AS Urlaubstage
      ,COUNT(Fz.Fz_ID) AS Urlaubseintraege
FROM Mitarbeiter AS Ma
LEFT JOIN Fehlzeit AS Fz ON Ma.Ma_ID = Fz.Ma_ID AND
                            Fz.FzA_ID = 2 AND
                            YEAR(Fz.VonDat) = 2022
LEFT JOIN Abteilung AS Abt ON Ma.Abt_ID = Abt.Abt_ID
GROUP BY Ma.Ma_ID, Ma.Nachname, Ma.Vorname, Abt.Abteilung;
```

b) 10 Punkte

```
SELECT Fz_ID
      ,Ma_ID
      ,FzA_ID
      ,VonDat
      ,BisDat
      ,FehltageSum
  INTO Fehlzeit_Archiv
  FROM Fehlzeit
  WHERE Ma_ID = 1;

DELETE
    FROM Fehlzeit
 WHERE Ma_ID = 1;
```

c) 5 Punkte

```
SELECT Ma.Ma_ID
      ,Ma.Nachname
      ,Ma.PLZ
      ,Ma.Ort
FROM Mitarbeiter AS Ma
WHERE Ma.PLZ LIKE '___3%'
```