

Familienname, Vorname (bitte durch eine Leerspalte trennen)

Bereich		Berufsnr.				IHK-Nr.			Prüflingsnr.			
6	7	1	2	0	1							
Sp. 1-2		Sp. 3-6				Sp. 7-9			Sp. 10-14			

Termin: Mittwoch, 4. Mai 2022



IHK

1201

2 Entwicklung und Umsetzung von Algorithmen

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung (AO 2020)

4 Aufgaben

mit Belegsatz

90 Minuten Prüfungszeit

100 Punkte

Hinweis:

Bei der Bearbeitung der Aufgaben ist von einem gewöhnlichen Geschäftsbetrieb auszugehen, der **nicht** durch die COVID-19-Pandemie beeinflusst bzw. durch entsprechende behördliche Verfügungen eingeschränkt ist.

Bearbeitungshinweise

1. Bevor Sie mit der Bearbeitung der Aufgaben beginnen, überprüfen Sie bitte die **Vollständigkeit** dieses Aufgabensatzes. Die Anzahl der zu bearbeitenden Aufgaben ist auf dem Deckblatt links angegeben. Wenden Sie sich bei Unstimmigkeiten sofort an die Aufsicht, weil Reklamationen am Ende der Prüfung nicht anerkannt werden können.
2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüfungs-Nr. in die oben stehenden Felder ein.
3. Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
4. Halten Sie sich bei der Bearbeitung der Aufgaben genau an die **Vorgaben der Aufgabenstellung** zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier Angaben bewertet.
5. Tragen Sie die frei zu formulierenden **Antworten dieser offenen Aufgaben** in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
6. Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
7. Schreiben Sie deutlich und gut lesbar. Ein nicht eindeutiges zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
8. Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
10. Für **Nebenrechnungen/Hilfsaufzeichnungen** können Sie das im Aufgabensatz enthaltene Konzeptpapier verwenden. Dieses muss vor der Bearbeitung der Aufgaben herausgetrennt werden. Bewertet werden jedoch nur Ihre Eintragungen im Aufgabensatz.

Wird vom Korrektor ausgefüllt!

Bewertung

Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen.

1. Aufg.

--	--

 Punkte 2. Aufg.

--	--

 Punkte 3. Aufg.

--	--

 Punkte 4. Aufg.

--	--

 Punkte

15 16 17 18 19 20 21 22

Prüfungszeit

23

Die entsprechende Ziffer (1, 2 oder 3) finden Sie in der Abfrage nach der Prüfungszeit im Anschluss an die letzte Aufgabe.

Gesamtpunktzahl

24	25	26

Prüfungsort, Datum

Unterschrift

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.

Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2022 – Alle Rechte vorbehalten!

Die Aufgaben 1 bis 4 beziehen sich auf die folgende Ausgangssituation:

Der Energieversorger Wind und Sonne AG möchte den Prozess zur Strom-Abrechnung weiter digitalisieren. Innerhalb dieses Projekts sollen mehrere Apps entwickelt werden. Dabei sollen Sie bei der Planung, Umsetzung und Einführung mitarbeiten und unterstützen.

Korrekturrand

1. Aufgabe (30 Punkte)

Für eine Statistik sollen die Energieverbräuche des jeweils letzten Jahres ausgewertet werden. Die Zählerstände zu jedem Monatsende eines Jahres liegen für alle Verbraucher in Zweipersonenhaushalten in einem zweidimensionalen Array `verbrauch` vor. In der Spalte 0 des Arrays ist die Verbrauchernummer und in der Spalte 1 der Zählerstand von Ende Dezember des Vorjahres eingetragen. Alle anderen Spalten enthalten die Zählerstände der einzelnen Monate des auszuwertenden Jahres. Die Zeile 0 enthält die Daten des ersten Verbrauchers.

Beispiel:

VerbraucherNr	Vorjahr	1	2	3	4	5	6	7	8	9	10	11	12
1001	23	28	31	35	42	45	47	50	56	61	67	73	78
1002	25	29	33	37	42	46	49	53	58	62	66	72	78
...													
5999	24	30	34	40	44	48	55	62	66	71	77	82	88

Für die Auswertung der Daten werden folgende statistische Werte benötigt:

- Minimaler Monatsverbrauch über alle Verbraucher hinweg
- Maximaler Monatsverbrauch über alle Verbraucher hinweg
- Für jeden Verbraucher, der in einem Monat einen vorgegebenen Schwellenwert (`limit`) überschritten hat, sind folgende Informationen in einem Objekt der Klasse `Monatsverbrauch` zu speichern:
die VerbraucherNr, die Monatsnummer (1 für Januar, ..., 12 für Dezember), der Verbrauch
Dieses Objekt wird anschließend einer Liste hinzugefügt.

Das zweidimensionale Array `verbrauch` steht innerhalb der Methode zur Verfügung, ebenso eine Variable `limit`, welche den vorgegebenen Schwellenwert enthält. Die Anzahl der Datensätze des Arrays kann durch `verbrauch.length` bestimmt werden. Für ein `List`-Objekt stehen folgende Methoden zur Verfügung:

<code>List()</code>	Standardkonstruktor
<code>add(Monatsverbrauch o)</code>	Hinzufügen eines Objektes

Erstellen Sie eine Methode `statistik(verbrauch: int[][], limit: int)` zur Auswertung der Verbrauchsdaten, welche alle geforderten Werte bestimmt.

Der Rückgabewert soll ein Objekt vom Typ `Jahresstatistik(minVerbrauch: int, maxVerbrauch: int, limitVerbraucher: List)` sein.

Geben Sie Ihre Lösung in Pseudocode, als Struktogramm, als Programmablaufplan oder als UML-Aktivitätsdiagramm an.

2. Aufgabe (20 Punkte)

a) Zum Testen der Methoden der zu entwickelnden Software kommen Unit-Tests zum Einsatz.

3 Punkte

aa) Beschreiben Sie, was ein Unit-Test ist und wozu er verwendet wird.

ab) Nennen und beschreiben Sie zwei Eigenschaften, welche ein Unit-Test erfüllen sollte.

4 Punkte

b) Im Rahmen des Projektes wurde eine Methode entwickelt, welche folgende funktionalen Anforderungen erfüllen soll:

- Die Methode soll die positive Differenz zwischen einem alten und einem neuen Zählerstand, welche als Parameter übergeben werden, berechnen und diesen Wert zurückgeben.
- Die Differenz soll nur dann berechnet werden, wenn die übergebenen Werte nicht negativ sind und der alte Zählerstand nicht größer als der neue Zählerstand ist. Ansonsten soll die Methode den Wert -1 zurückgeben.

Quellcode der Funktion:

```
int berechneDifferenz (int zaehlerstandAlt, int zaehlerstandNeu) {  
    int differenz = -1;  
    if (zaehlerstandAlt >= 0 && zaehlerstandNeu > 0) {  
        if (zaehlerstandAlt < zaehlerstandNeu) {  
            differenz = zaehlerstandNeu - zaehlerstandAlt;  
        }  
    }  
    return differenz;  
}
```

ba) Zum Testen der Methode wurde folgender Unittest entwickelt.

Korrekturrand

```
@Test
void testeBerechneZahlungsdifferenz () {
    assertEquals(20, berechneDifferenz(240, 260)); // Testfall 1
    assertEquals(0, berechneDifferenz(120, 120)); // Testfall 2
    assertEquals(0, berechneDifferenz(0, 0)); // Testfall 3
    assertEquals(80, berechneDifferenz(0, 80)); // Testfall 4
    assertEquals(-1, berechneDifferenz(440, 420)); // Testfall 5
}
```

assertEquals(Expected, Actual)	Prüft, ob die Werte für Actual und Expected gleich sind. Ist dieses nicht der Fall, schlägt der Test fehl.
--------------------------------	---

Überprüfen Sie, ob der Unit-Test erfolgreich ist oder fehlschlägt. Tragen Sie dazu das Ergebnis von jedem Testfall in die Tabelle ein. (OK – wenn der Test erfolgreich ist, Fehler – wenn der Test fehlschlägt.) 5 Punkte

Testfall	Ergebnis des Tests
1	
2	
3	
4	
5	

bb) Der Unit-Test schlägt fehl.

Passen Sie den Quellcode so an, dass der Unit-Test erfolgreich durchlaufen wird.

4 Punkte

```
int berechneDifferenz (int zaehlerstandAlt, int zaehlerstandNeu) {

    int differenz = -1;

    if (zaehlerstandAlt >= 0 && zaehlerstandNeu > 0) {

        if (zaehlerstandAlt < zaehlerstandNeu) {
            differenz = zaehlerstandNeu - zaehlerstandAlt;
        }

    }

    return differenz;
}
```

bc) Im Unit-Test fehlen noch wichtige Testfälle.

Geben Sie Testdaten aus zwei weiteren Äquivalenzklassen an, für welche die Methode den Rückgabewert -1 liefern sollte.

4 Punkte

Nr.	zaehlerstandAlt	zaehlerstandNeu	differenz
1			
2			

3. Aufgabe (20 Punkte)

Die notwendigen Daten für eine Tarifübersicht sollen in einer relationalen Datenbank erfasst werden. Eine Tarifübersicht für ein Jahr sieht wie folgt aus:

Tarifübersicht

Kundennummer: 1234

Kunde:

Maier, Clara
Waldessaum 11
99999 Tannen

Ausstellungsdatum: 21.04.2022

Ansprechpartner:

Müller, Hans
Telefon: 0111- 45678
Mail: hans.mueller@energieversorger.de

Sehr geehrter Kunde,

im Jahr 2021 haben Sie folgende Tarife gebucht:

Ökostrom normal: vom 01.01.2021 bis 03.04.2021
Ökostrom 3+: vom 04.04.2021 bis 31.10.2021
Ökostrom normal: vom 01.11.2021 bis 31.12.2021

Ihr Energieversorger

Hinweise: Der in der Übersicht genannte Ansprechpartner ist der persönliche Ansprechpartner des Kunden beim Energieversorger. Das Ausstellungsdatum wird in der Datenbank nicht erfasst.

Erstellen Sie das entsprechende Datenbankmodell in der 3. Normalform.

- Geben Sie alle erforderlichen Tabellen und die entsprechenden Attribute an.
- Kennzeichnen Sie Primärschlüssel mit PK und Fremdschlüssel mit FK.
- Zeichnen Sie die Beziehungen zwischen den Tabellen mit deren Kardinalitäten ein.

4. Aufgabe (30 Punkte)

Für die Verwaltung und Auswertung der Daten soll die Datenbankabfragesprache SQL zum Einsatz kommen.

a) In einer Teambesprechung wurde beschlossen, dass u. a. auch Stored Procedures, Trigger und Indizes verwendet werden.

aa) Beschreiben Sie, was Sie unter dem Begriff „Stored Procedure“ verstehen.

2 Punkte

ab) Beschreiben Sie, was ein Trigger ist.

2 Punkte

ac) Beschreiben Sie, wozu Spalten in Tabellen indiziert werden.

2 Punkte

b) Im Zuge der Datenbankentwicklung wurden u. a. folgende Tabellen entwickelt.

Haushalte

HH_ID	HH_Nachname	HH_Vorname	HH_StrasseHausnr	HH_PLZ	HH_Ort
1	Meier	Manni	Hauptstr. 5	50667	Köln
2	Hinz	Heiner	Bahnhofstr. 233	50679	Köln
3	Müller	Petra	Baumgasse 1	50667	Köln
4	Conrad	Heinz	Treppentsteg 2	50667	Köln

Zaehler

Z_ID	Z_HHID	Z_Nummer
1	2	0037890053
2	2	0037889999
3	1	1666856771
4	3	5658897888
5	1	8885688977

Zaehlerstand

ZST_ID	ZST_ZID	ZST_Datum	ZST_Stand
1	1	07.01.2021	450003
2	1	05.01.2022	458977
3	3	12.01.2020	453
4	3	13.02.2021	1988
5	3	04.01.2022	4022
6	2	12.01.2020	1588
7	2	13.02.2021	4298
8	2	04.01.2022	6999
9	5	13.01.2019	1
10	5	21.01.2020	1866
11	5	04.01.2021	3755
12	5	13.01.2022	6500

Zur Abfrage und Pflege der Daten sollen nachfolgende SQL-Anweisungen erstellt werden.

ba) Indizieren Sie in der Tabelle *Zaehlerstand* das Feld *ZSt_Datum*. Der Index soll die Bezeichnung *Idx_Datum* bekommen.

3 Punkte

bb) Löschen Sie alle Haushalte, bei denen kein Zähler hinterlegt wurde.

5 Punkte

b) Erstellen Sie eine SQL-Abfrage, mit der Sie alle Haushalte mit der Anzahl der vorhandenen Zähler erhalten. Haushalte mit den wenigsten Zählern sollen unten aufgeführt werden.

6 Punkte

Hinweis: Beachten Sie die Spaltenbezeichnungen.

Beispiel-Ergebnistabelle:

Nachname ▾	AnzZaehler ▾
Meier	2
Hinz	2
Müller	1
Conrad	0

10 Punkte

Beispiel-Ergebnistabelle:

[illegible]