

# 1 Ganzheitliche Aufgabe | Fachqualifikationen

5 Handlungsschritte  
90 Minuten Prüfungszeit  
100 Punkte

## Bearbeitungshinweise

1. Der vorliegende Aufgabensatz besteht aus insgesamt 5 Handlungsschritten zu je 25 Punkten.  
**In der Prüfung zu bearbeiten sind 4 Handlungsschritte, die vom Prüfungsteilnehmer frei gewählt werden können.**  
Der nicht bearbeitete Handlungsschritt ist durch Streichung des Aufgabentextes im Aufgabensatz und unten mit dem Vermerk „Nicht bearbeiteter Handlungsschritt: Nr. ... „ an Stelle einer Lösungsniederschrift deutlich zu kennzeichnen. Erfolgt eine solche Kennzeichnung nicht oder nicht eindeutig, gilt der 5. Handlungsschritt als nicht bearbeitet.
2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-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 Aufgabenstellungen** 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 eindeutig 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. Ein **Tabellenbuch** oder ein **IT-Handbuch** oder eine **Formelsammlung** ist als Hilfsmittel zugelassen.
11. Für **Nebenrechnungen/Hilfsaufzeichnungen** können Sie das im Aufgabensatz enthaltene Konzeptpapier verwenden. Dieses muss vor Bearbeitung der Aufgaben herausgetrennt werden. Bewertet werden jedoch nur Ihre Eintragungen im Aufgabensatz.

**Nicht bearbeiteter Handlungsschritt ist Nr.**

**Wird vom Korrektor ausgefüllt!**

## Bewertung

**Bewertung**  
Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen. Für den abgewählten Handlungsschritt ist anstatt der Punktzahl die Buchstabenkombination „AA“ in die Kästchen einzutragen.

Spalte 1 - 14  
s. o.

Punkte 1. Handlungsschritt 15 16

Punkte 2. Handlungsschritt 17 18

Punkte 3. Handlungsschritt 19 20

Punkte 4. Handlungsschritt 21 22

Punkte 5. Handlungsschritt 23 24

Gesamtpunktzahl

26 27 28

Prüfungsort Datum

Prüfungszeit

25

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

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 2012 – Alle Rechte vorbehalten!

**Situation**

Sie sind Mitarbeiter/-in des IT-Dienstleiters Immo-IT GmbH, die sich auf die Entwicklung von Software im Bereich des Immobilienmanagement und der Vermarktung spezialisiert hat. Sie wurde von der Bengel&Gölp GmbH (B&G GmbH), einem internationalen Maklerbüro mit mehreren Niederlassungen im In- und Ausland, mit der Restrukturierung des IT-Systems beauftragt.

Im Rahmen dieses Auftrags sollen Sie vier der folgenden fünf Aufgaben erledigen:

1. Ein UML-Anwendungsfalldiagramm und eine Methode erstellen
2. Ein UML-Zustandsdiagramm und Klassendiagramm erstellen
3. Ein relationales Datenbankmodell entwickeln
4. Einen Algorithmus entwickeln
5. SQL-Anweisungen erstellen

**1. Handlungsschritt (25 Punkte)**

Die B&G GmbH vermietet auch exklusive Ferienhäuser. Die Immo-IT GmbH soll dazu eine Anwendung entwickeln, über die im Internet Ferienhäuser angeboten und gebucht werden können.

a) Das System soll Folgendes ermöglichen:

- Ein Vermieter stellt ein Ferienhaus ein.
- Ein Kunde sucht ein Ferienhaus.
- Ein Kunde prüft die Verfügbarkeit eines ausgewählten Ferienhauses.
- Ein Kunde stellt einen Reservierungsauftrag und gibt alle erforderlichen Daten ein.
- Ein Vermieter reserviert ein Ferienhaus und verschickt eine Reservierungsbestätigung an den Kunden.

Erstellen Sie ein UML-Anwendungsfalldiagramm.

(12 Punkte)



b) Für das neue Online-Buchungssystem der B&G GmbH ist eine Methode „getHolidayEstates( )“ mit folgender Funktionalität zu erstellen:

- Ferienhäuser ermitteln, die den übergebenen Suchkriterien entsprechen
- Rückgabe des Suchergebnisses als Liste der gefundenen Objekte
- Entspricht kein Ferienhaus den Suchkriterien, soll eine entsprechende Meldung ausgegeben und die Methode beendet werden.

Der Methode werden dazu folgende Suchkriterien (Parameter) übergeben:

- Region, in der das gesuchte Ferienhaus liegen soll (destination)
- Anzahl Personen, für die das Ferienhaus mindestens ausgelegt ist (persons)
- Anzahl Schlafzimmer, die das Ferienhaus mindestens haben soll (bedrooms)
- Mietpreis pro Tag, der höchstens verlangt werden darf (maxPrice)
- Datum des Anreisetag (arrival)
- Mietdauer in Tagen (duration)

Die Funktion greift hierzu auf eine Liste aller im Buchungssystem erfassten Ferienhäuser der B&G GmbH zu.

Folgende Funktionalitäten wurden bereits erstellt und stehen zur Verfügung:

Name	Übergabeparameter	Rückgabewert	Beschreibung
getEstates( )	destination: String	Estate[ ]	Liefert eine Liste aller Objekte <i>am Wunschort</i> („ <i>destination</i> “) vom Typ Estate
getBedrooms( )	estate: Estate	Integer	Liefern die entsprechenden Eigenschaften eines <i>Estate</i> -Objektes
getPersons( )	estate: Estate	Integer	
getPrice( )	estate: Estate	Double	
getVacancies()	arrival: Date duration: Integer estate: Estate	Boolean	
createList()		List	Erstellt ein Objekt vom Typ List, in der beliebige Objekte gespeichert werden können
add( )	Estate	-	Methode der Klasse List; fügt ein neues Objekt in die Liste ein

Stellen Sie die Logik in Pseudocode, in einem Struktogramm oder PAP dar.

(13 Punkte)



## 2. Handlungsschritt (25 Punkte)

Korrekturrand

Die B&G GmbH verkauft Immobilien.

- a) Eine Verkaufsimmobilie kann die folgenden Zustände haben:
- Wenn sie zum Verkauf freigegeben wurde, steht sie zur Verfügung.
  - Für zur Verfügung stehende Immobilien können Anfragen entgegengenommen werden. Die Immobilie ist damit angefragt.
  - Für angefragte Immobilien können weitere Anfragen erfolgen.
  - Zur Verfügung stehende oder angefragte Immobilien können reserviert werden (nur **eine** Reservierung ist möglich). Die Immobilie ist dann reserviert.
  - Reservierte Immobilien können verkauft werden.
  - Mit dem Unterschreiben des Verkaufsvertrages wird der Endzustand erreicht.

Hinweis:

- Auf jede Anfrage wird Informationsmaterial verschickt.
- Anfragen bleiben bei einer Reservierung bestehen.
- Bei einer Reservierung wird eine Reservierungsbestätigung verschickt.
- Reservierungen können zurückgenommen werden.

Erstellen Sie ein UML-Zustandsdiagramm.

(13 Punkte)

- b) Vervollständigen Sie zu den nachstehend beschriebenen Beziehungen die folgende Tabelle entsprechend der Symbolik eines UML-Klassendiagramms (ohne Attribute und Kardinalitäten). Geben Sie dazu den Namen der Beziehung und eine Begründung an, warum diese Beziehung von Ihnen gewählt wurde. (12 Punkte)

Korrekturrand

Beschreibung	Beziehungstyp	Klassendiagramm	Begründung
Eine Immobilie besteht aus mehreren Wohnungen.			
Bewohner können entweder Mieter oder Eigentümer sein.			
In einer Mietervereinigung gibt es mehrere Mieter.			

### 3. Handlungsschritt (25 Punkte)

Korrekturrand

Die B&G GmbH hat bisher folgende Daten mit einem Tabellenkalkulationsprogramm verwaltet (siehe nebenstehende Anlage).

Die Immo-IT GmbH wurde beauftragt, zur Verwaltung dieser Daten eine Datenbank zu entwickeln.

Erstellen Sie dazu ein relationales Datenmodell in der dritten Normalform. Kennzeichnen Sie Primärschlüssel mit PK und Fremdschlüssel mit FK.

Fortsetzung 3. Handlungsschritt →



Anlage zum 3. Handlungsschritt, Tabelle der B&G GmbH

Immobilie	Adresse der Immobilie	Eigentümer	Adresse des Eigentümers	Makler	Telefon	Kunde	Adresse des Kunden	Besuchstermin
A la cart Hotel	Am Leuchtturm 12, 80333 München	Bella Hotel AG	Am Hafen, 20144 Hamburg	Berta Maier	089 1234-43	Hans Müller	Schulstr. 45, 93055 Regensburg	13.01.2012
A la cart Hotel	Am Leuchtturm 12, 80333 München	Bella Hotel AG	Am Hafen, 20144 Hamburg	Klaus Baldus	089 1234-12	Schöller GmbH	Im Rhein 2, 50999 Köln	02.02.2012
Villa Hügel	Kruppstr. 1, 45130 Essen	Alfredo Kruppi	Kruppstr. 1, 45130 Essen	Klaus Baldus	089 1234-12	Gerd Hoesch	Hochofen Allee 12, 44787 Bochum	14.02.2012
Filou Theater	Babelsberg Str. 20, 14480 Potsdam	Marlene Schulz	Kudamm 23, 10115 Berlin	Toni Sauer	089 1234-40	Gerd Hoesch	Hochofen Allee 12, 44787 Bochum	15.02.2012
Hotel Penthaus	Schreiberweg 132, 80335 München	Bella Hotel AG	Am Hafen, 20144 Hamburg	Klaus Baldus	089 1234-12	Schöller GmbH	Im Rhein 2, 50999 Köln	02.02.2012

#### 4. Handlungsschritt (25 Punkte)

Korrekturrand

Die B&G GmbH führt im Rahmen der Hausverwaltung Nebenkostenabrechnungen durch.

Die Immo-IT GmbH soll eine Funktion erstellen, mit der die Gesamtnebenkosten für jedes Haus ermittelt werden können.

Die einzelnen Nebenkosten liegen in einer XML-Datei vor.

Beispiel

```
<Nebenkosten>
  <Haus ID="1234">
    <Muell >1.200,00</Muell>
    <Wasser>3.000,00</Wasser>
    <Strom>500,00</Strom>
    <Versicherungen>2400,00</Versicherungen>
  </Haus>
  <Haus ID="4321">
    <Muell >3.800,00</Muell>
    <Wasser>8.900,00</Wasser>
    <Strom>1.200,00</Strom>
    <Versicherungen>7.800,00</Versicherungen>
  </Haus>
  <!-- Nebenkosten für weitere Häuser -->
</Nebenkosten>
```

Es wird eine Funktion `Nebenkosten()` benötigt, die für jedes Haus die Nebenkostenpositionen ausgibt und die Summe aller Nebenkosten für jedes Haus berechnet und ausgibt.

Der Funktion `Nebenkosten()` soll als Parameter eine Referenz auf ein Objekt vom Typ `Element` übergeben werden.

Über diese Referenz kann der Wurzelknoten `<Nebenkosten>` bearbeitet werden.

Ihnen stehen folgende Methoden zur Verfügung:

Methodenname	Beschreibung
<code>getElementsByTagName</code> (String:knotenname) : NodeList	Methode der Klasse <code>Element</code> : liefert Referenzen auf alle Objekte, mit denen die Knoten <code>&lt;knotenname&gt;</code> bearbeitet werden können in einem Objekt vom Typ <code>NodeList</code> Bsp: <code>NodeList liste = wurzel.getElementsByTagName("Haus")</code>
<code>getLength()</code> : Integer	Methode der Klasse <code>NodeList</code> : Liefert die Anzahl der Elemente in dieser <code>NodeList</code>
<code>item(integer: i)</code> : Node	Methode der Klasse <code>NodeList</code> : Liefert eine Referenz auf das i-te Element der <code>NodeList</code> vom Typ <code>Node</code>
<code>getAttributes()</code> : <code>NamedNodeMap</code>	Methode der Klasse <code>Node</code> : liefert Referenzen auf Objekte, mit denen die Attribute dieses Knotens bearbeitet werden können Bsp.: <code>NamedNodeMap map = node.getAttributes()</code>
<code>item(integer)</code> : Node	Methode der Klasse <code>NamedNodeMap</code> : Liefert eine Referenz auf das i-te Element der <code>NamedNodeMap</code> vom Typ <code>Node</code>
<code>getNodeValue()</code> : String	Methode der Klasse <code>Node</code> : Liefert den Knotenwert als Text Hinweis: Die angegebenen Zahlenwerte sind <b>immer</b> die Knotenwerte des ersten <b>Kindknotens</b> der Knoten <code>&lt;Muell&gt;</code> , <code>&lt;Wasser&gt;</code> , <code>&lt;Strom&gt;</code> bzw. <code>&lt;Versicherungen&gt;</code>
<code>getNodeName()</code> : String	Methode der Klasse <code>Node</code> : Liefert den Namen des Knotens
<code>getChildNodes()</code> : NodeList	Methode der Klasse <code>Node</code> : Liefert Referenzen auf Objekte, mit denen die Kindknoten dieses Knotens bearbeitet werden können Bsp.: <code>NodeList list = node.getChildNodes()</code>
<code>getFirstChild()</code> : Node	Methode der Klasse <code>Node</code> : Liefert eine Referenz auf ein Objekt, mit der der erste Kindknoten dieses Knotens bearbeitet werden kann

### Beispielausgabe:

Korrekturrand

```
Haus: H1234  
Muell: 1200.00  
Wasser: 3000.00  
Strom: 500.00  
Versicherungen: 2400.00  
Haus-Gesamtkosten: 7100.00
```

```
Haus: H4321  
Muell: 3800.00  
Wasser: 8900.00  
Strom: 1200.00  
Versicherungen: 7800.00  
Haus-Gesamtkosten: 21700.00
```

Erstellen Sie die Logik zur Funktion `Nebenkosten()`. Sie soll für jedes Haus die Nebenkostenpositionen auflisten und die Summe aller Nebenkosten je Haus berechnen und ausgeben.

Stellen Sie die Logik in Pseudocode, in einem Struktogramm oder PAP dar.

18.1.2020

18.1.2020

## Korrekturrand

## Tabellen

<b>Ferienhaus</b>
Ferienhaus_ID
Eigentuemer_ID
Adresse_ID
AnzahlSchlafzimmer

Eigentuemmer
Eigentuemmer_ID
Adresse_ID
Name

<b>Mietvertrag</b>
Mietvertrag_ID
Ferienhaus_ID
Kunde_ID
Beginn
Ende
Tage

<b>Maengelanzeige</b>
Maengelanzeige_ID
Ferienhaus_ID
Meldedatum
Beschreibung

Kunde
Kunde_ID
Adresse_ID
Name

Adresse
Adresse_ID
Adresse
...

a) Erstellen Sie eine SQL-Anweisung, mit der Sie das Ferienhaus mit der höchsten Auslastung (Anzahl Tage) erhalten. (5 Punkte)

[illegible][illegible]

- c) Erstellen Sie eine SQL-Anweisung, mit der Sie alle Ferienhäuser mit Mängelanzeigen im Verhältnis zu der Anzahl der Buchungen aufsteigend sortiert nach dem Verhältnis erhalten. (5 Punkte)

- d) Erstellen Sie eine SQL-Anweisung, mit der Sie eine Liste aller Ferienhaus-IDs nach Anzahl der Vermietungstage absteigend sortiert erhalten. (5 Punkte)

- e) Erstellen Sie eine SQL-Anweisung, mit der Sie alle Ferienhäuser, die eine unterdurchschnittliche Auslastung (unter 50 %) haben, nach Auslastung absteigend sortiert erhalten. (5 Punkte)

bitte wenden!

**PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!**

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

- ☐ 1 Sie hätte kürzer sein können.    ☐ 2 Sie war angemessen.    ☐ 3 Sie hätte länger sein müssen.

☐



