



Name: _____

Beispielaufgabe

Informatik, Grundkurs

Aufgabenstellung

Für den neugegründeten Online-Sportschuh-Versand „RunningShoe“ soll eine Software zur Verwaltung ihrer Kunden entwickelt werden.

Abbildung 1 zeigt einen Ausschnitt des Implementationsdiagramms. Ein Auszug aus der Dokumentation ist im Anhang zu finden.

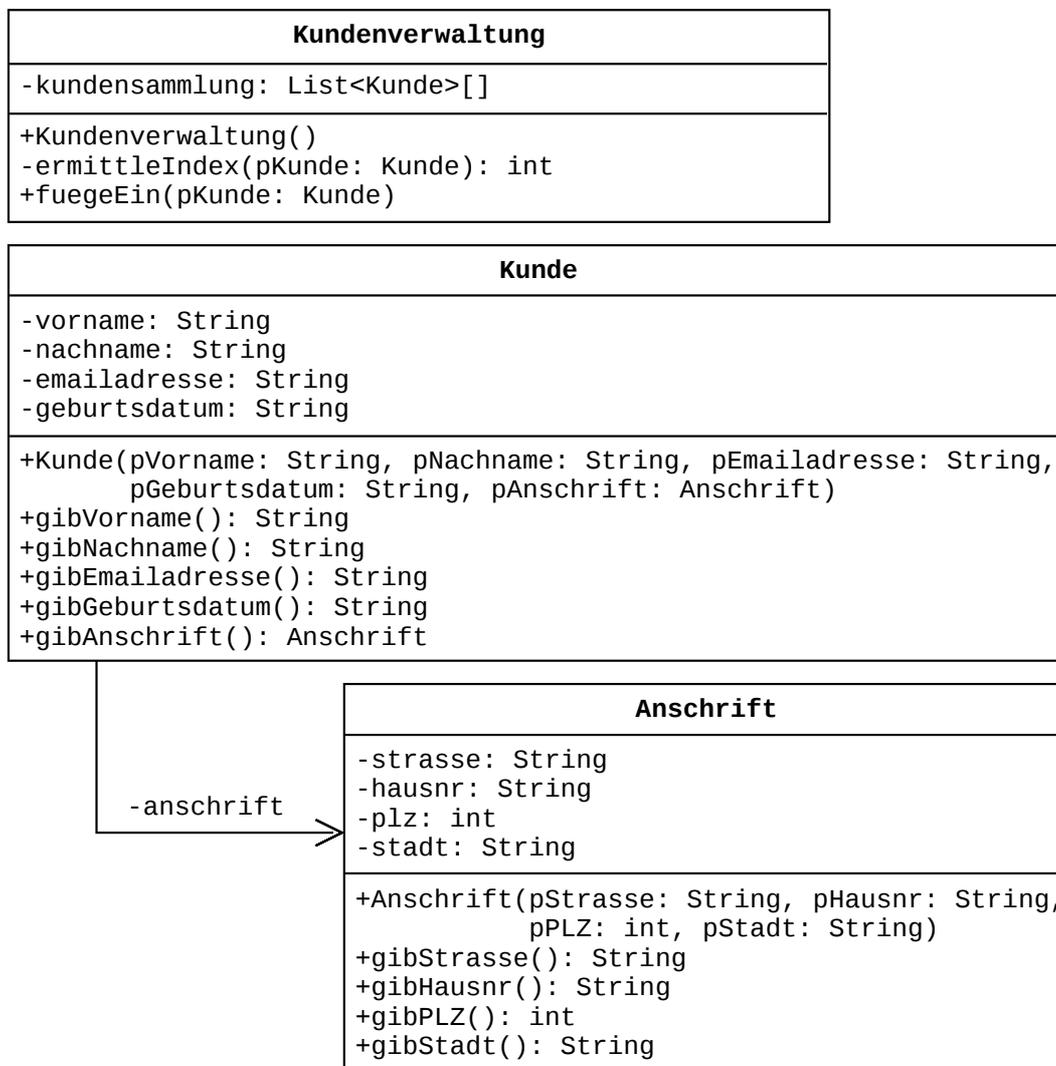


Abbildung 1: Implementationsdiagramm (Teilmodellierung)



Name: _____

Ein neuer Kunde wird in die Kundenverwaltung in Abhängigkeit der ersten zwei Ziffern von der Postleitzahl seiner Anschrift in eine von 100 Listen mit Objekten der Klasse Kunde eingefügt.

Die jeweilige Liste steht dabei im Array kundensammlung unter dem Index, der den ersten beiden Ziffern der PLZ entspricht.

Beispielsweise wird ein Kunde aus Münster (PLZ: 48143) in die Liste mit dem Index 48 und ein Kunde aus Leipzig (PLZ: 04103) in die Liste mit dem Index 4 des Arrays kundensammlung eingefügt.

a) Folgende vier Kunden werden in die Kundenverwaltung eingefügt:

- i. Emma Meier, geboren am 02.03.1970,
mit der E-Mailadresse emma@meier.de
und der Anschrift Bahnhofstr. 1, 44135 Dortmund.
- ii. Luise Schumann, geboren am 24.10.1982,
mit der E-Mailadresse luise@schumann.de
und der Anschrift Schulstr. 23, 80331 München.
- iii. Paul Krause, geboren am 16.06.1965,
mit der E-Mailadresse paul@krause.de
und der Anschrift Am See 80, 01067 Dresden.
- iv. Thorsten Müller, geboren am 30.03.1990,
mit der E-Mailadresse thorsten@mueller.de
und der Anschrift Bahnhofstr. 2, 33615 Bielefeld.

Geben Sie an, unter welchem Index im Array kundensammlung die vier Datensätze eingefügt werden.

Erläutern Sie die Beziehungen zwischen den Klassen Kundenverwaltung, Kunde und Anschrift.

(5 Punkte)



Name: _____

- b) In der Klasse Kundenverwaltung ist die Methode fuegeEin wie folgt implementiert:

```
1 public void fuegeEin(Kunde pKunde) {
2     String neuerNachname = pKunde.gibNachname();
3     String neuerVorname = pKunde.gibVorname();
4     List<Kunde> aktuelleListe =
5         kundensammlung[ermittleIndex(pKunde)];
6     aktuelleListe.moveToFirst();
7     boolean positionGefunden = false;
8     while (aktuelleListe.hasAccess() && !positionGefunden) {
9         Kunde aktuellerKunde = aktuelleListe.getContent();
10        String aktuellerNachname = aktuellerKunde.gibNachname();
11        String aktuellerVorname = aktuellerKunde.gibVorname();
12        if ((aktuellerNachname.compareTo(neuerNachname) < 0)
13            || (aktuellerNachname.equals(neuerNachname)
14                && aktuellerVorname.compareTo(neuerVorname) < 0)) {
15            aktuelleListe.next();
16        } else {
17            positionGefunden = true;
18        }
19    }
20    if (positionGefunden) {
21        aktuelleListe.insert(pKunde);
22    } else {
23        aktuelleListe.append(pKunde);
24    }
25 }
```

Analysieren Sie die Methode und erläutern Sie die Funktionsweise.

Erläutern Sie die Bedeutung des Werts der Variablen positionGefunden.

(11 Punkte)



Name: _____

- c) Die Geschäftsführung möchte zu statistischen Zwecken herausfinden, welche Kunden in einem Ort mit einer bestimmten PLZ wohnen.

Die Methode `ermittleKunden` wurde zu diesem Zweck in die Klasse `Kundenverwaltung` eingefügt. Sie ist wie folgt dokumentiert:

`public List<Kunde> ermittleKunden(int pPLZ)`

Eine Liste mit den Objekten vom Typ `Kunde` wird zurückgeliefert, deren PLZ identisch mit der im Parameter übergebenen ist.

Falls es keinen Kunden mit der im Parameter übergebenen PLZ gibt, wird eine leere Liste zurückgeliefert.

Entwickeln Sie einen Algorithmus für die Arbeitsweise der Methode `ermittleKunden` und stellen Sie diese in geeigneter Form dar.

Implementieren Sie die Methode `ermittleKunden`.

Erläutern Sie unter Berücksichtigung des Implementationsdiagramms, warum alle Kunden mit einer bestimmten Postleitzahl im Allgemeinen schneller ermittelt werden können, als alle Kunden mit einem bestimmten Geburtstag.

(15 Punkte)

- d) Alle Kunden hatten bei der Anmeldung für den Online-Sportschuh-Versand bestätigt, dass ihre Kundendaten für die mit dem Kaufprozess relevanten Zwecke benutzt werden dürfen.

Der Geschäftsführer des Online-Sportschuh-Versands „RunningShoe“ überlegt, die Kundendaten mit einem befreundeten Geschäftsführer, der Outdoor-Sportkleidung vertreibt, auszutauschen, um den Kunden das bestmögliche Einkaufserlebnis zu ermöglichen.

Beurteilen Sie auf Grundlage der Grundprinzipien des Datenschutzes die Idee des Geschäftsführers und erläutern Sie dabei exemplarisch ein wesentliches Grundprinzip, auf das Sie sich beziehen.

(6 Punkte)



Name: _____

e) Über die Kundenverwaltung sollen nun auch die Bestellungen der einzelnen Kunden verwaltet werden.

In der Modellerweiterung müssen folgende Anforderungen berücksichtigt werden:

- i. Für jeden Kunden müssen seine Bestellungen mit Bestellposten verwaltet werden.
- ii. Ein Kunde muss neue Bestellungen aufgeben können.
Die Informationen über all seine Bestellungen müssen abgefragt werden können.
- iii. Eine Bestellung besteht aus einer Bestellnummer und beliebig vielen Bestellposten. Die Informationen über die Bestellnummer und über die Bestellposten müssen auch abgefragt werden können.
Außerdem muss man einer Bestellung weitere Bestellposten hinzufügen können.
- iv. Für einen Bestellposten werden lediglich die Artikelnummer, die Artikelbezeichnung, der Stückpreis und die bestellte Anzahl verwaltet.
Diese Informationen müssen auch abgerufen werden können.

Modellieren Sie die oben genannten Anforderungen als Erweiterung des Implementationsdiagramms aus Abbildung 1.

Hinweis: Unveränderte Attribute, Methoden und Assoziationen aus dem Implementationsdiagramm (Abbildung 1) müssen nicht aufgeführt werden.

Erläutern Sie, wie Sie jede der genannten Anforderungen in Ihrem Implementationsdiagramm realisieren.

(13 Punkte)

Zugelassene Hilfsmittel:

- Wörterbuch zur deutschen Rechtschreibung
- Taschenrechner (grafikfähiger Taschenrechner / CAS-Rechner)



Name: _____

Anhang

Die Klasse Kundenverwaltung

Objekte der Klasse **Kundenverwaltung** verwalten beliebig viele Objekte vom Typ Kunde gruppiert nach der Postleitzahl.

Dokumentation der Klasse Kundenverwaltung

Kundenverwaltung()

Ein Objekt dieser Klasse wird initialisiert.

private int ermittleIndex(Kunde pKunde)

Die Methode liefert in Abhängigkeit von der PLZ des Kunden den Index im Array zurück, der den ersten beiden Ziffern der Postleitzahl entspricht. Falls pKunde null ist oder die Postleitzahl nicht existiert, wird (-1) zurückgeliefert.

public fuegeEin(Kunde pKunde)

(vgl. Teilaufgabe b.)



Name: _____

Die generische Klasse `List<ContentType>`

Objekte der generischen Klasse `List` verwalten beliebig viele, linear angeordnete Objekte vom Typ `ContentType`. Auf höchstens ein Listenobjekt, aktuelles Objekt genannt, kann jeweils zugegriffen werden. Wenn eine Liste leer ist, vollständig durchlaufen wurde oder das aktuelle Objekt am Ende der Liste gelöscht wurde, gibt es kein aktuelles Objekt. Das erste oder das letzte Objekt einer Liste können durch einen Auftrag zum aktuellen Objekt gemacht werden. Außerdem kann das dem aktuellen Objekt folgende Listenobjekt zum neuen aktuellen Objekt werden.

Das aktuelle Objekt kann gelesen, verändert oder gelöscht werden. Außerdem kann vor dem aktuellen Objekt ein Listenobjekt eingefügt oder ein Listenobjekt an das Ende der Liste angefügt werden.

Dokumentation der Klasse `List`

`List<ContentType>()`

Eine leere Liste wird erzeugt.

`boolean isEmpty()`

Die Anfrage liefert den Wert `true`, wenn die Liste keine Objekte enthält, sonst liefert sie den Wert `false`.

`boolean hasAccess()`

Die Anfrage liefert den Wert `true`, wenn es ein aktuelles Objekt gibt, sonst liefert sie den Wert `false`.

`void next()`

Falls die Liste nicht leer ist, es ein aktuelles Objekt gibt und dieses nicht das letzte Objekt der Liste ist, wird das dem aktuellen Objekt in der Liste folgende Objekt zum aktuellen Objekt, andernfalls gibt es nach Ausführung des Auftrags kein aktuelles Objekt, d. h., `hasAccess()` liefert den Wert `false`.

`void toFirst()`

Falls die Liste nicht leer ist, wird das erste Objekt der Liste aktuelles Objekt. Ist die Liste leer, geschieht nichts.

`void toLast()`

Falls die Liste nicht leer ist, wird das letzte Objekt der Liste aktuelles Objekt. Ist die Liste leer, geschieht nichts.



Name: _____

ContentType getContent()

Falls es ein aktuelles Objekt gibt (`hasAccess() == true`), wird das aktuelle Objekt zurückgegeben, andernfalls (`hasAccess() == false`) gibt die Anfrage den Wert `null` zurück.

void setContent(ContentType pContent)

Falls es ein aktuelles Objekt gibt (`hasAccess() == true`) und `pContent` ungleich `null` ist, wird das aktuelle Objekt durch `pContent` ersetzt. Sonst bleibt die Liste unverändert.

void append(ContentType pContent)

Ein neues Objekt `pContent` wird am Ende der Liste eingefügt. Das aktuelle Objekt bleibt unverändert. Wenn die Liste leer ist, wird das Objekt `pContent` in die Liste eingefügt und es gibt weiterhin kein aktuelles Objekt (`hasAccess() == false`). Falls `pContent` gleich `null` ist, bleibt die Liste unverändert.

void insert(ContentType pContent)

Falls es ein aktuelles Objekt gibt (`hasAccess() == true`), wird ein neues Objekt vor dem aktuellen Objekt in die Liste eingefügt. Das aktuelle Objekt bleibt unverändert. Falls die Liste leer ist und es somit kein aktuelles Objekt gibt (`hasAccess() == false`), wird `pContent` in die Liste eingefügt und es gibt weiterhin kein aktuelles Objekt. Falls es kein aktuelles Objekt gibt (`hasAccess() == false`) und die Liste nicht leer ist oder `pContent` gleich `null` ist, bleibt die Liste unverändert.

void concat(List pList)

Die Liste `pList` wird an die Liste angehängt. Anschließend wird `pList` eine leere Liste. Das aktuelle Objekt bleibt unverändert. Falls `pList` `null` oder eine leere Liste ist, bleibt die Liste unverändert.

void remove()

Falls es ein aktuelles Objekt gibt (`hasAccess() == true`), wird das aktuelle Objekt gelöscht und das Objekt hinter dem gelöschten Objekt wird zum aktuellen Objekt. Wird das Objekt, das am Ende der Liste steht, gelöscht, gibt es kein aktuelles Objekt mehr (`hasAccess() == false`). Wenn die Liste leer ist oder es kein aktuelles Objekt gibt (`hasAccess() == false`), bleibt die Liste unverändert.

Unterlagen für die Lehrkraft

Beispielaufgabe

Informatik, Grundkurs

1. Aufgabenart

Analyse, Modellierung und Implementation von kontextbezogenen Problemstellungen mit Schwerpunkt auf den Inhaltsfeldern Daten und ihre Strukturierung, Algorithmen und Informatik, Mensch und Gesellschaft

2. Aufgabenstellung

siehe Prüfungsaufgabe

3. Materialgrundlage

entfällt

4. Bezüge zu dem Kernlehrplan und den Vorgaben

Die Aufgaben weisen vielfältige Bezüge zu den Kompetenzerwartungen und Inhaltsfeldern des Kernlehrplans bzw. zu den in den Vorgaben ausgewiesenen Fokussierungen auf. Im Folgenden wird auf Bezüge von zentraler Bedeutung hingewiesen.

1. Inhaltsfelder und inhaltliche Schwerpunkte

Daten und ihre Strukturierung

- Objekte und Klassen
 - Implementationsdiagramme
 - Lineare Strukturen (Array, lineare Liste)

Algorithmen

- Analyse, Entwurf und Implementierung von Algorithmen
- Algorithmen in ausgewählten informatischen Kontexten

Formale Sprachen und Automaten

- Syntax und Semantik einer Programmiersprache
 - Java

Informatik, Mensch und Gesellschaft

- Wirkungen der Automatisierung
 - Grundprinzipien des Datenschutzes

2. Medien/Materialien

entfällt

5. Zugelassene Hilfsmittel

- Wörterbuch zur deutschen Rechtschreibung
- GTR (graphikfähiger Taschenrechner) oder CAS (Computer-Algebra-System)

6.1 Vorbemerkung zu dieser Beispielklausur

Das Ziel dieser Beispielaufgabe ist es, eine mögliche Verknüpfung der Inhaltsfelder „Daten und ihre Strukturierung“, „Algorithmen“ und „Informatik, Mensch und Gesellschaft“ für eine inhaltsfeldübergreifende Grundkursklausur zu zeigen.

Diese Beispielaufgabe orientiert sich an den Abiturvorgaben für 2021.

Ab dem Abiturjahrgang 2021 ist für eine Grundkursklausur, die aus zwei Aufgaben besteht, eine Bearbeitungszeit von 3 Stunden und 45 Minuten vorgesehen. Diese Beispielaufgabe stellt nur eine von zwei Aufgaben dar, die zusammen eine vollständige Grundkursklausur ergeben.

6.2 Modelllösungen

Die Modelllösung stellt eine mögliche Lösung bzw. Lösungsskizze dar. Der gewählte Lösungsansatz und –weg der Schülerinnen und Schüler muss nicht identisch mit dem der Modelllösung sein. Sachlich richtige Alternativen werden mit entsprechender Punktzahl bewertet (Bewertungsbogen: Zeile „Sachlich richtige Lösungsalternative zur Modelllösung“).

Teilaufgabe a)

- i. Der Datensatz für Emma Meier wird in die Liste unter dem Index 44 des Arrays kundensammlung eingefügt.
- ii. Der Datensatz für Luise Schumann wird in die Liste unter dem Index 80 im Array kundensammlung eingefügt.
- iii. Der Datensatz für Paul Krause wird in die Liste unter dem Index 1 im Array kundensammlung eingefügt.
- iv. Der Datensatz für Thorsten Müller wird in die Liste unter dem Index 33 im Array kundensammlung eingefügt.

Ein Objekt der Klasse Kundenverwaltung verwaltet Listen mit Objekten vom Typ Kunde in einem eindimensionalen Array kundensammlung. Es können somit beliebig viele Kundenobjekte verwaltet werden.

Ein Objekt vom Typ Kunde verwaltet ein Objekt anschrift vom Typ Anschrift.

Teilaufgabe b)

Auf Grundlage der Postleitzahl des neu einzufügenden Kunden wird die entsprechende Liste im eindimensionalen Array bestimmt. Diese Liste wird durch die lokale Variable `aktuelleListe` referenziert (vgl. Zeile 4).

Die `aktuelleListe` wird von vorne durchlaufen (vgl. Zeilen 5ff.) und die richtige Einfügestelle wird gesucht. D. h. es wird das Element gesucht, vor das eingefügt werden soll.

Solange das Listenende noch nicht erreicht ist und die Einfügeposition noch nicht gefunden ist, wird die Referenz auf das aktuelle Listenelement auf das Folgeelement weiterbewegt, wenn der Nachname des aktuellen Elements in der lexikografischen Ordnung vor dem neu einzufügenden Nachnamen steht oder die Nachnamen identisch sind und der Vorname des

aktuellen Elements in der lexikografischen Sortierreihenfolge vor dem neu einzufügenden Vornamen steht (vgl. Zeilen 11f.). Sonst ist die Einfügestelle gefunden (vgl. Zeile 14).

Wenn die Einfügestelle gefunden wurde, wird die neue Person eingefügt, also vor der ersten Person mit lexikografisch folgendem Namen, und sonst ans Listenende angehängt (vgl. Zeilen 17-21).

Die Methode fügt ein neues Kundenobjekt sortiert nach Nachnamen und Vornamen in eine Liste von Kunden ein. Die entsprechende Liste aus dem eindimensionalen Array wird auf Grundlage der Postleitzahl ermittelt.

Die boolesche Variable `positionGefunden` gibt an, ob eine Einfügestelle (zum Einfügen vor einem bestehenden Kunden) gefunden wurde.

Die Variable `positionGefunden` wird in der Schleife auf `true` gesetzt, sobald ein Datensatz in der Liste gefunden wurde, vor dem das neue Kundenobjekt eingefügt werden muss.

Nach Durchlauf der Schleife gibt die Variable an, ob an der aktuellen Stelle der Liste ein Kunde eingefügt werden soll (`true`) oder der neue Kunde an die Liste angehängt wird, weil keine Einfügeposition gefunden wurde (`false`).

Teilaufgabe c)

Eine neue leere Liste für die gesuchten Kunden wird erzeugt.

Die Liste an der Position `pPLZ/1000` der `kundensammlung` wird komplett durchlaufen.

Die Postleitzahl des aktuellen Listenobjekts wird jeweils mit der im Parameter übergebenen Postleitzahl verglichen. Wenn die Postleitzahlen übereinstimmen, dann wird das aktuelle Listenobjekt an die neue Liste mit den gesuchten Kunden angehängt.

Die neue Liste mit den gesuchten Kunden wird zurückgeliefert.

```
public List<Kunde> ermittleKunden(int pPLZ) {
    List<Kunde> gesuchteKunden = new List<Kunde>();
    List<Kunde> kundenliste = kundensammlung[pPLZ/1000];
    kundenliste.moveToFirst();
    while(kundenliste.hasNext()) {
        if(kundenliste.getContent().gibAnschrift().gibPLZ() == pPLZ) {
            gesuchteKunden.append(kundenliste.getContent());
        }
        kundenliste.next();
    }
    return gesuchteKunden;
}
```

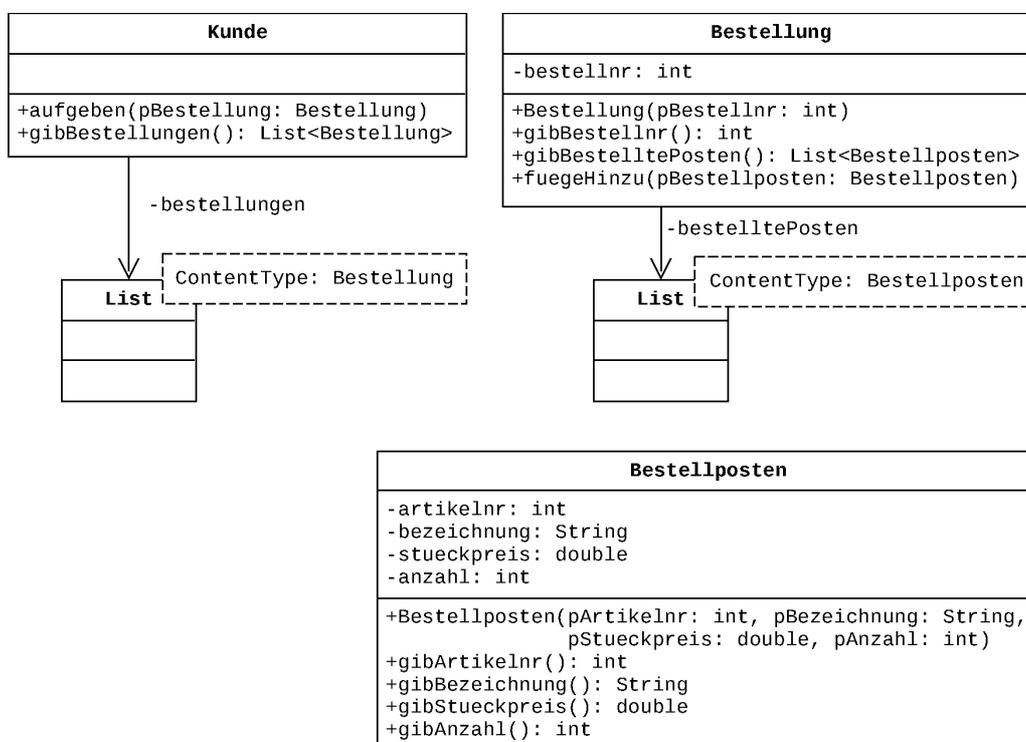
Die Kunden mit einer bestimmten Postleitzahl können schneller ermittelt werden, weil die Kundenobjekte gruppiert nach Postleitzahlbereichen verwaltet werden. Wenn eine Person mit einer bestimmten Postleitzahl gesucht wird, müssen nicht alle Kundenobjekte durchlaufen werden, sondern nur die Kunden im gleichen Postleitzahlbereich. Da es 100 Postleitzahlbereiche gibt (von 0 bis 99), müssen bei einer gleichmäßigen Aufteilung der Kunden auf die Postleitzahlbereiche, nur ca. 1/100 der gesamten Kunden durchlaufen werden.

Da die Kundenobjekte nicht nach Geburtsdatum gruppiert oder sortiert sind, müssen im Gegensatz dazu alle Kundenobjekte durchlaufen werden.

Teilaufgabe d)

Die Idee des Geschäftsführers, die Kundendaten mit einem befreundeten Geschäftsführer auszutauschen bzw. an ihn weiterzugeben, ist – vor dem Hintergrund der Grundprinzipien des Datenschutzes – abzulehnen. Der Kunde hat nur der Verarbeitung seiner Kundendaten für die mit dem Kaufprozess relevanten Zwecke zugestimmt. Da der Kunde der Weitergabe seiner Daten vorher nicht explizit zugestimmt hat, widerspricht dieses Vorhaben dem Prinzip des Verbots mit Erlaubnisvorbehalt.

Unter dem Prinzip des Verbots mit Erlaubnisvorbehalt wird verstanden, dass die Verarbeitung, d. h. zum Beispiel die Erhebung, Speicherung, Weitergabe oder allgemeine Verwendung personenbezogener Daten grundsätzlich verboten ist – es sei denn, die betroffene Person hat der Verarbeitung für einen konkreten Zweck zugestimmt oder es gibt eine explizite gesetzliche Regelung, die eine Verarbeitung für einen konkreten Zweck erlaubt.

Teilaufgabe e)

- Zu i) Ein Kunde verwaltet eine Liste bestellungen mit Objekten vom Typ Bestellung. Eine Bestellung verwaltet eine Liste bestelltePosten mit Objekten vom Typ Bestellposten.
- Zu ii) Die Klasse Kunde enthält die Methoden aufgeben und gibBestellungen.
- Zu iii) Eine Bestellung verwaltet die bestellnr und eine Liste bestelltePosten mit Objekten vom Typ Bestellposten. Die Getter-Methoden gibBestellnr und gibBestelltePosten liefern die entsprechenden Informationen zurück. Mit der Methode fuegeHinzu können weitere Bestellposten einer Bestellung hinzugefügt werden.
- Zu iv) Ein Bestellposten verwaltet die artikelnr, die bezeichnung, den stueckPreis und die anzahl. Die Getter-Methoden liefern die entsprechenden Informationen zurück.

7. Teilleistungen – Kriterien / Bewertungsbogen zur Prüfungsarbeit

Name des Prüflings: _____ Kursbezeichnung: _____

Schule: _____

Teilaufgabe a)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK ¹	ZK	DK
1	gibt an, an unter welchem Index im Array kundensammlung die vier Datensätze eingefügt werden.	2 (I)			
2	erläutert die Beziehungen zwischen den Klassen Kundenverwaltung, Kunde und Anschrift.	3 (I-II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (5)					
Summe Teilaufgabe a)		5			

Teilaufgabe b)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	analysiert die Methode und erläutert die Funktionsweise.	8 (II)			
2	erläutert die Bedeutung des Werts der Variablen positionGefunden.	3 (I-II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (11)					
Summe Teilaufgabe b)		11			

¹ EK = Erstkorrektur; ZK = Zweitkorrektur; DK = Drittkorrektur

Teilaufgabe c)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	entwickelt einen Algorithmus für die Arbeitsweise der Methode ermittelte Kunden und stellt diesen in geeigneter Form dar.	5 (II)			
2	implementiert die Methode ermittelte Kunden.	7 (II)			
3	erläutert unter Berücksichtigung des Implementationsdiagramms, warum alle Kunden mit einer bestimmten Postleitzahl im Allgemeinen schneller ermittelt werden können, als alle Kunden mit einem bestimmten Geburtstag.	3 (II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (15)					
Summe Teilaufgabe c)		15			

Teilaufgabe d)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	beurteilt – auf Grundlage der Grundprinzipien des Datenschutzes – die Idee des Geschäftsführers.	3 (III)			
2	erläutert dabei exemplarisch ein wesentliches Grundprinzip, auf das er bzw. sie sich bezieht.	3 (I-II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (6)					
Summe Teilaufgabe d)		6			

Teilaufgabe e)

Anforderungen		Lösungsqualität			
Der Prüfling		maximal erreichbare Punktzahl (AFB)	EK	ZK	DK
1	modelliert die Erweiterung als Implementationsdiagramm.	8 (II-III)			
2	erläutert, wie jede Anforderung im Implementationsdiagramm realisiert wird.	5 (I-II)			
Sachlich richtige Lösungsalternative zur Modelllösung: (13)					
Summe Teilaufgabe e)		13			