

Abiturprüfung 2016

INFORMATIK

Arbeitszeit: 180 Minuten

Der Fachausschuss wählt je eine Aufgabe aus den Gebieten
Inf1 und Inf2 zur Bearbeitung aus.

Der Fachausschuss ergänzt im folgenden Feld die erlaubten
objektorientierten Programmiersprachen:

INF1. MODELLIERUNG UND PROGRAMMIERUNG

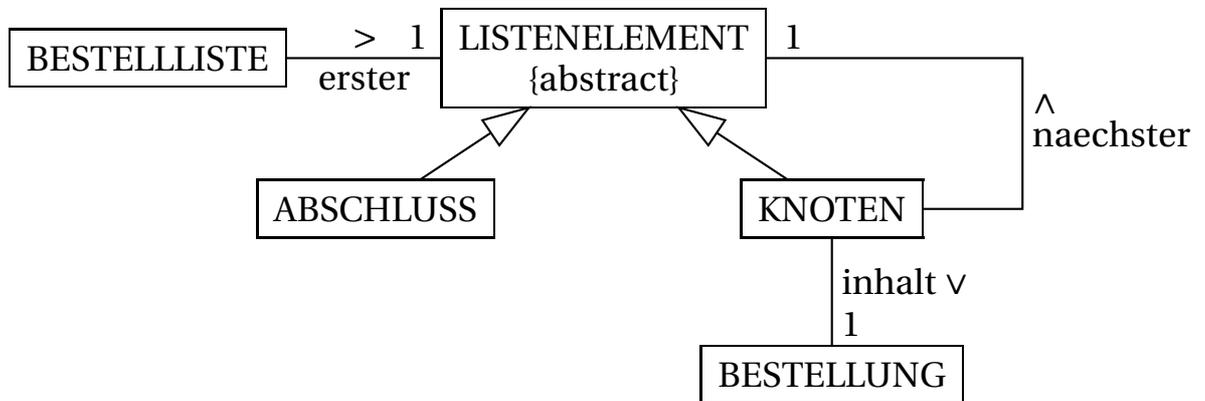
I.

BE	
	<p>Die Firma MuesLi will europaweit individuelle Müslimischungen vertreiben und lässt dazu eine passende Software erstellen.</p>
4	<p>1. Nennen und beschreiben Sie kurz vier typische Phasen bei der Durchführung eines Softwareprojekts.</p> <p>Die Software soll folgende Anforderungen erfüllen: Ein Kunde, der sich auf der Homepage der Firma MuesLi mit Vorname, Nachname, Lieferadresse und E-Mail-Adresse anmeldet, kann sich aus verschiedenen Zutaten persönliche Müslimischungen zusammenstellen und diese bestellen. Eine Mischung wird festgelegt durch die gewünschte Gesamtmenge in Gramm und den Anteil jeder Zutat in Prozent. Vereinfachend wird angenommen, dass für eine Müslimischung nur folgende sechs Zutaten zur Verfügung stehen: Bananen, Dinkelflocken, Haferflocken, Mandeln, Rosinen und Schokolade. Jeder Bestellung werden der Kunde, die zusammengestellten Müslimischungen und das Bestelldatum zugeordnet. Eine gelungene Mischung kann der Kunde auch veröffentlichen, so dass weitere Kunden diese Mischung bestellen können.</p> <p>9 2. Erstellen Sie ein Klassendiagramm für das beschriebene Szenario unter Verwendung der Klassen BESTELLSYSTEM, KUNDE, BESTELLUNG und MUESLIMISCHUNG.</p>

(Fortsetzung nächste Seite)

BE

3. Die bei der Firma MuesLi eingehenden Bestellungen sollen in der Reihenfolge ihres Eingangs bearbeitet werden. Hierfür werden die Bestellungen in einer einfach verketteten Liste gemäß dem folgenden Klassendiagramm verwaltet:



- 3 a) Die Software-Entwickler hätten die Verwaltung der Bestellungen auch mit einem Feld umsetzen können. Nehmen Sie begründet Stellung, ob hier die Verwendung eines Feldes auch sinnvoll wäre.
- 4 b) Erstellen Sie für die verkettete Liste ein Objektdiagramm, das zwei Bestellungen enthält, eine vom 29.04.2016 und eine vom 03.05.2016. Beschränken Sie sich auf Objekte der im vorgegebenen Klassendiagramm angegebenen Klassen.
- 8 c) Die Methode *umsatzBerechnen* der Klasse **BESTELLLISTE** gibt die Summe der Gesamtpreise aller in der Liste vorhandenen Bestellungen zurück. Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache diese Methode und alle dazu benötigten Methoden der übrigen Klassen unter Beachtung des Prinzips der Rekursion. Die Klasse **BESTELLUNG** kann als bereits implementiert vorausgesetzt werden; insbesondere existiert eine Methode *preisBerechnen*, die den Gesamtpreis der Bestellung zurückgibt.
- 6 d) Stellen Sie den Ablauf beim Aufruf der Methode *umsatzBerechnen* für die in Teilaufgabe 3b erstellte Liste graphisch dar, z. B. in einem Sequenzdiagramm.

(Fortsetzung nächste Seite)

BE

4. Die Kundendaten sollen mithilfe eines geordneten Binärbaums verwaltet werden. Die Daten werden alphabetisch nach der E-Mail-Adresse sortiert eingeordnet.

Hinweis: Zur Bezeichnung jedes Knotens genügt es im Folgenden, die den Kunden eindeutig identifizierende E-Mail-Adresse anzugeben.

4

- a) Die Daten der Kunden mit den E-Mail-Adressen

fischer@wep.com,
krause@wep.de,
leicht@gmy.com,
meier@gmy.de,
meyer@gmy.de,
schulze@aoe.de und
stark@gmy.de

wurden so in einen geordneten Binärbaum eingefügt, dass dieser möglichst wenige Ebenen besitzt. Stellen Sie einen solchen Baum dar und geben Sie zudem eine mögliche zugehörige Einfügereihenfolge an.

4

- b) Die Geschäftsleitung der Firma MuesLi rechnet mit einer großen Zahl von Kunden.

Schätzen Sie für 10 000 Kunden die Anzahl der Vergleiche ab, die maximal bei einem Suchvorgang benötigt werden, wenn der Baum möglichst wenige Ebenen hat.

8

- c) Der Baum ist unter Verwendung des Softwaremusters Kompositum implementiert. Dazu kann das Klassendiagramm aus Aufgabe 3 wiederverwendet werden, wenn u. a. BESTELLUNG durch KUNDE ersetzt wird. Geben Sie an, wie sich – abgesehen von Bezeichnern – das Klassendiagramm der Baumstruktur vom Klassendiagramm in Aufgabe 3 unterscheidet.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache alle Methoden, die für die Ausgabe der E-Mail-Adressen aller Kunden in alphabetischer Reihenfolge benötigt werden.

Sie können dazu die Methode *emailadresseAusgeben* der Klasse KUNDE als bereits implementiert voraussetzen.

(Fortsetzung nächste Seite)

BE

5. Während der Zusammenstellung seiner Müslimischung sollen einem Kunden weitere Zutaten zu seiner aktuell ausgewählten Zutat vorgeschlagen werden. Hat der Kunde z. B. als aktuelle Zutat Haferflocken ausgewählt, sollen ihm die Zutaten empfohlen werden, die bereits andere Kunden mit Haferflocken zusammengestellt haben.

Um diese Empfehlungen geben zu können, müssen die Daten über die Zusammenstellungen gespeichert werden.

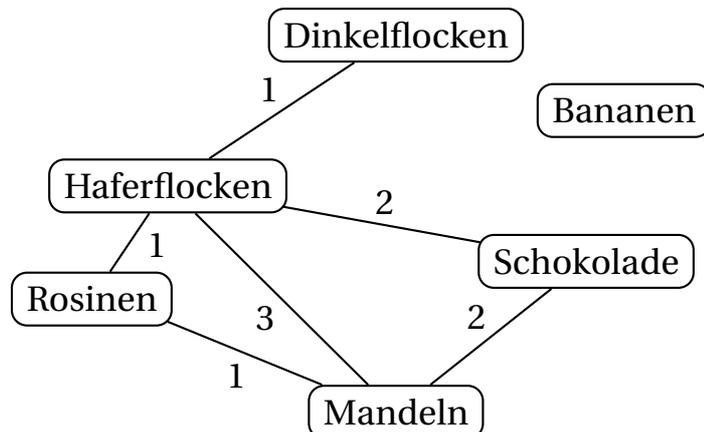
Folgende vier Zutatenkombinationen wurden im abgebildeten Graphen ausgewertet.

Kombination A: Haferflocken, Rosinen und Mandeln

Kombination B: Haferflocken, Mandeln und Schokolade

Kombination C: Dinkelflocken und Haferflocken

Kombination D: Haferflocken, Schokolade und Mandeln



Die Zahl an jeder Kante gibt an, in wie vielen Kombinationen beide Zutaten enthalten sind. Ist keine Kante vorhanden, trat diese Zutatenkombination noch nicht auf.

6

a) Nennen Sie zwei wesentliche Eigenschaften dieses Graphen. Repräsentieren Sie den gegebenen Graphen durch eine Adjazenzmatrix.

In der Klasse MUESLIMATRIX soll der Graph auf der Basis einer Adjazenzmatrix implementiert sein. Die Bezeichner der Knoten sollen ebenfalls in dieser Klasse in einem Feld von Zeichenketten gespeichert werden.

(Fortsetzung nächste Seite)

BE	
6	<p>b) Geben Sie für die Klasse MUESLIMATRIX in einer auf dem Deckblatt angegebenen Programmiersprache die Deklaration der Attribute sowie eine Implementierung des Konstruktors an. Im Konstruktor werden den Knoten die Zutatenbezeichner Bananen, Dinkelflocken, Haferflocken, Mandeln, Rosinen und Schokolade zugeordnet; ferner wird eine Adjazenzmatrix erstellt und deren Einträge explizit mit der Zahl 0 besetzt.</p>
5	<p>c) Die Methode <i>dazuEmpfehlen(i)</i> soll aus der Adjazenzmatrix auslesen, welche Zutaten bereits mit der aktuellen Zutat, die im Knotenfeld den Index <i>i</i> hat, mehr als einmal kombiniert wurden und diese dem Kunden als Empfehlung anzeigen. Für den obigen Graphen soll für die Zutat „Mandeln“ beispielhaft folgendes ausgegeben werden: „Viele Kunden haben zu Ihrer Zutat noch Haferflocken gewählt. Viele Kunden haben zu Ihrer Zutat noch Schokolade gewählt.“ Stellen Sie den Algorithmus für die Umsetzung der Methode <i>dazuEmpfehlen(i)</i> graphisch dar, z. B. durch ein Struktogramm.</p>
8	<p>d) Für die Entwicklung neuer Müslimischungen ist es hilfreich zu wissen, welche Zutaten auch indirekt über andere Zutaten miteinander in Beziehung stehen (im obigen Graphen z. B. Dinkelflocken und Rosinen). Formulieren Sie unter Bezug auf die Adjazenzmatrix einen Algorithmus, mit dem ausgehend von einem Startknoten alle erreichbaren Knoten genau einmal ausgegeben werden.</p>
	<p>Informationen über die Zutaten werden dauerhaft in einer Datenbanktabelle <i>zutaten</i> gespeichert. Neben dem Bezeichner (<i>bezeichner</i>) sind unter anderem die vorrätige Menge (<i>menge</i>), der aus Erfahrungswerten bekannte Verbrauch pro Monat (<i>verbrauch</i>) und der Preis pro Kilogramm (<i>preis</i>) gespeichert.</p>
3	<p>e) Zur Neige gehende Zutaten müssen nachbestellt werden. Geben Sie eine Datenbankabfrage (z. B. in SQL) an, die die Bezeichner all der Zutaten ausgibt, bei denen die vorhandene Menge kleiner als der aus Erfahrungswerten bekannte Monatsverbrauch ist.</p>
2	<p>f) Geben Sie an, welche Schritte in einem objektorientierten Programm zur Durchführung einer Datenbankabfrage prinzipiell notwendig sind.</p>
80	

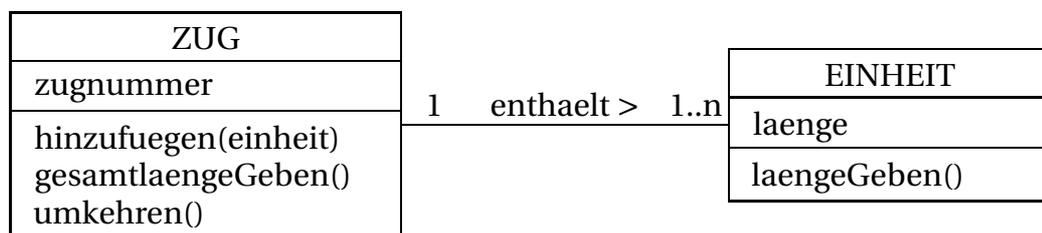
INF1. MODELLIERUNG UND PROGRAMMIERUNG

II.

BE

Ein kleines Eisenbahnunternehmen betreibt Sonderzüge mit historischen Lokomotiven und Wagen.

1. Zur Planung der Zusammenstellung eines Sonderzuges kommt eine auf dem folgenden Klassendiagramm basierende Software zum Einsatz:



Eine Einheit kann Lokomotive, Personenwagen oder Gepäckwagen sein.

4

- a) Für die verschiedenen Arten von Einheiten werden weitere Informationen gespeichert: bei Lokomotiven die Leistung, bei Personenwagen die Anzahl der Sitzplätze und bei Gepäckwagen die zulässige Lademasse. Ergänzen Sie das gegebene Klassendiagramm so, dass es der beschriebenen Situation Rechnung trägt.

12

- b) Die Klasse ZUG verwaltet maximal 20 Einheiten mithilfe eines Feldes. In ihr werden u. a. folgende Methoden vereinbart:

- Der Konstruktor, dem die Zugnummer und die Lokomotive als Parameter übergeben werden, erstellt ein Objekt, das nur die Lokomotive als erstes Feldelement enthält.
- Die Methode *hinzufuegen(einheit)* fügt den als Parameter übergebenen Wagen am Ende des Zuges hinzu. Es darf vorausgesetzt werden, dass keine Lokomotiven übergeben werden.
- Die Methode *umkehren()* kehrt die Reihenfolge der Wagen des Zuges um und belässt die Lokomotive an erster Stelle.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung der Klasse ZUG nach diesen Vorgaben. Sie können davon ausgehen, dass die Klasse EINHEIT bereits vollständig implementiert ist.

(Fortsetzung nächste Seite)

BE

2. Für jede Sonderzugfahrt sind die Bahnhöfe, beginnend mit 0, der Reihe nach vom Startbahnhof bis zum Zielbahnhof durchnummeriert.

Die Mitfahrt in den Zügen ist nur nach vorheriger Buchung und nur für registrierte Kunden und deren Mitreisende möglich; es können auch Teilstrecken gebucht werden.

Bei einer solchen Buchung auf der Website des Unternehmens werden die Kundennummer des Kunden, die Fahrtnummer der Zugfahrt, die Anzahl der Personen, die Bahnhofnummer des Einstiegsbahnhofs und die Bahnhofnummer des Ausstiegsbahnhofs erfasst und in einem Objekt der Klasse BUCHUNG gespeichert.

Die für eine Zugfahrt vorliegenden Buchungen werden in einer einfach verketteten Liste der Klasse BUCHUNGSLISTE abgespeichert. Dabei kommt das Entwurfsmuster Kompositum unter Berücksichtigung des Prinzips der Trennung von Struktur und Daten zum Einsatz.

5 a) Stellen Sie eine solche Liste mit drei Objekten der Klasse BUCHUNG in einem Objektdiagramm dar. Auf die Angabe von Attributen und Attributwerten in den Objekten der Klasse BUCHUNG kann dabei verzichtet werden.

12 b) Bei der Planung einer Zugfahrt ist es wichtig, für die einzelnen Teilstrecken der Zugfahrt die jeweilige Anzahl der Reisenden abrufen zu können.

Die Methode *anzahlReisendeGeben(bahnhofsnummer)* der Klasse BUCHUNGSLISTE gibt die Anzahl derjenigen Personen zurück, deren gebuchte Fahrt die Teilstrecke zwischen dem durch den Parameter festgelegten Bahnhof und dem darauf folgenden Bahnhof beinhaltet.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache unter Beachtung des Prinzips der Rekursion eine Implementierung dieser Methode sowie der dafür benötigten Methoden in den anderen Klassen.

Die folgenden Attribute der Klasse BUCHUNG können dabei als gegeben vorausgesetzt werden:

- *personenanzahl* enthält die Anzahl der Personen dieser Buchung,
- *bahnhof1* enthält die Nummer des Einstiegsbahnhofs,
- *bahnhof2* enthält die Nummer des Ausstiegsbahnhofs.

BE

2

c) Im Anschluss an die eigentliche Buchung können auf Wunsch auch Sitzplätze reserviert werden. In einer graphischen Übersicht ist dabei für jeden einzelnen Sitzplatz erkennbar, ob er bereits reserviert oder noch verfügbar ist.

Zur Speicherung des Reservierungszustandes hat man sich für den Einsatz zweier verschiedener Datenstrukturen entschieden: Die einzelnen Personenwagen des Zuges werden hier in einer verketteten Liste verwaltet, die Sitzplätze jedes einzelnen Personenwagens hingegen in einem Feld.

Geben Sie mögliche Gründe für diese Entscheidungen an.

8

d) Insgesamt sind auf der Website des Unternehmens fünf Seiten für die Abwicklung von Buchungen vorgesehen:

- Auf der Startseite wird der Buchungsvorgang erklärt.
Durch Drücken auf „Weiter“ gelangt man zur eigentlichen Buchungsseite.
- Auf der Buchungsseite gibt man die Daten der Buchung ein.
Durch einen Klick auf „Platzreservierung“ werden die Daten gespeichert und man gelangt zur Reservierungsseite; klickt man dagegen auf „Weiter“, werden die Daten gespeichert und man gelangt direkt zur Kontrollseite.
- Auf der Reservierungsseite kann man Sitzplätze reservieren.
Durch eine Bestätigung mit „Weiter“ werden die Reservierungsdaten gespeichert und man gelangt zur Kontrollseite.
- Auf der Kontrollseite werden noch einmal alle Eingaben angezeigt und man wird dazu aufgefordert, in die allgemeinen Geschäftsbedingungen des Unternehmens einzuwilligen.
Durch einen Klick auf „Abbrechen“ werden die gespeicherten Daten gelöscht und man gelangt zurück zur Startseite.
Falls man in die allgemeinen Geschäftsbedingungen eingewilligt hat, wird durch den Schalter „Kostenpflichtig buchen“ die Buchung durchgeführt und man gelangt zur Quittungsseite.
- Auf der Quittungsseite bestätigt das Unternehmen die Buchung.
Mit dem Schalter „OK“ gelangt man wieder zur Startseite.

Stellen Sie diese Abläufe in einem Zustandsübergangdiagramm dar.

BE

3. Die Daten aller Kunden des Unternehmens werden in einer Datenbank gespeichert.

Von Zeit zu Zeit kommt es vor, dass einzelne Kunden ihre Kundennummer vergessen, die für die Anmeldung auf der Website erforderlich ist. In solchen Fällen rufen die betroffenen Kunden bei der Hotline des Unternehmens an, nennen ihren Vornamen und Nachnamen und bekommen daraufhin ihre Kundennummer noch einmal genannt.

3 a) Nachdem der Kunde Herbert Huber seine Kundennummer vergessen hat, soll aus der Tabelle *kunden* eine Liste der Kundennummern aller Kunden mit dem Vornamen Herbert und dem Nachnamen Huber abgerufen werden.

Geben Sie eine entsprechende Datenbankabfrage (z. B. in SQL) an.

3 b) Nennen Sie ein Argument, inwiefern eine Auskunft der beschriebenen Art unter dem Gesichtspunkt des Datenschutzes kritisch ist und geben Sie an, wie durch eine Änderung des Verfahrens gegebenenfalls eine Verbesserung erzielt werden könnte.

4 c) Zur Ermöglichung schneller Suchvorgänge legen viele Datenbanken die gespeicherten Daten ganz oder teilweise in Datenstrukturen ab, die einem geordneten Binärbaum ähneln.

Zeichnen Sie einen nach Nachnamen geordneten Binärbaum, in den in der angegebenen Reihenfolge nacheinander Personen mit den folgenden Nachnamen aufgenommen worden sind:

Adenauer, Erhard, Kiesinger, Brandt, Schmidt, Kohl, Schröder, Merkel.

6 d) Bestimmen Sie die Anzahl der Ebenen, die ein geordneter Binärbaum mit zwei Millionen Knoten im günstigsten und im ungünstigsten Fall hat. Erklären Sie ausgehend von diesen beiden Zahlenwerten kurz, weshalb ein geordneter Binärbaum nicht in allen Fällen eine optimale Suchgeschwindigkeit sicherstellt.

(Fortsetzung nächste Seite)

BE

4. Die Sonderfahrten finden unter anderem auf den folgenden Eisenbahnstrecken statt:

Augsburg \rightleftarrows Donauwörth

Augsburg \rightleftarrows Ingolstadt

Donauwörth \rightleftarrows Treuchtlingen

Ingolstadt \rightleftarrows Treuchtlingen

Treuchtlingen \rightleftarrows Nürnberg

Treuchtlingen \rightleftarrows Ansbach

Ansbach \rightleftarrows Würzburg

Nürnberg \rightleftarrows Würzburg

Würzburg \rightleftarrows Frankfurt

Zur Speicherung dieses Streckennetzes soll die Datenstruktur Graph verwendet werden.

2

a) Erläutern Sie kurz, weshalb auf der Basis der bisherigen Angaben lediglich ein ungerichteter und unbewerteter Graph erstellt werden kann.

6

b) Zeichnen Sie einen derartigen Graphen des Streckennetzes und geben Sie seine Adjazenzmatrix an.

3

c) Im Rahmen einer Jubiläumsfahrt sollen alle genannten Bahnhöfe des Streckennetzes angefahren werden.

Nennen Sie einen Algorithmus zum Graphendurchlauf und erläutern Sie, inwieweit er für die Lösung dieser Aufgabe geeignet ist.

(Fortsetzung nächste Seite)

BE

10

d) Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung der Klasse STRECKENGRAPH, in welcher der in Teilaufgabe 4b beschriebene Graph abgespeichert werden kann. Bahnhofsnamen und Adjazenzmatrix sollen in geeigneten Feldern gespeichert werden. Es kann davon ausgegangen werden, dass niemals mehr als 100 Bahnhöfe verwaltet werden müssen. Beschränken Sie Ihre Implementierung auf Attributdeklarationen, einen geeigneten Konstruktor sowie die folgenden beiden Methoden:

- Die Methode *bahnhofHinzufuegen(bahnhofsname)* fügt einen neuen Bahnhof mit dem als Parameter übergebenen Namen hinzu.
- Die Methode *streckeHinzufuegen(bahnhofsname1, bahnhofsname2)* fügt eine neue Strecke zwischen den beiden bereits bestehenden Bahnhöfen mit den als Parameter übergebenen Namen hinzu.

Eine Methode *bahnhofsnummerGeben(bahnhofsname)*, welche den Feldindex des Bahnhofs mit dem als Parameter übergebenen Namen zurückgibt, kann dabei als gegeben vorausgesetzt werden.

Auf mögliche fehlerhafte Eingaben, wie beispielsweise doppelt vorkommende oder nicht existierende Bahnhofsnamen, muss nicht eingegangen werden.

80

INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

III.

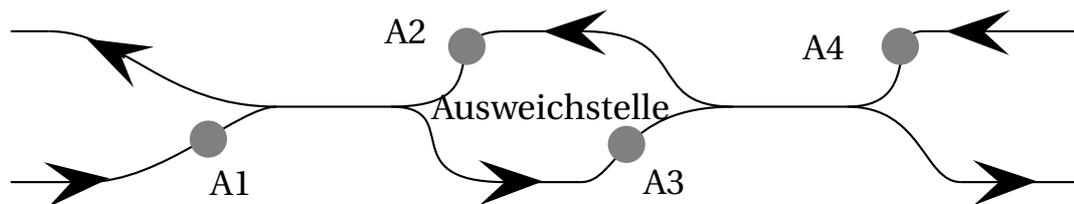
BE

1. Über das Online-System einer Konzertagentur können Plätze für Konzerte reserviert werden. Jeder Nutzer des Reservierungssystems kann sich dort informieren, wie viele Plätze für ein bestimmtes Konzert noch vorhanden sind, und dann gegebenenfalls Karten reservieren.

5 a) Erläutern Sie unter Zuhilfenahme einer geeigneten graphischen Darstellung (z. B. eines Sequenzdiagramms), wie es unter bestimmten Umständen zu einer fehlerhaften Reservierung kommen kann.

3 b) Beschreiben Sie ein Konzept, mit dem die Zugriffe auf gemeinsame Ressourcen synchronisiert werden können. Erläutern Sie, wie dieses Konzept angewendet werden kann, um den in Teilaufgabe 1a beschriebenen Fehler zu verhindern.

4 2. Eine zweispurige Straße für zwei Richtungen ist im Baustellenbereich auf eine Spur verengt. In der Mitte des Baustellenbereichs wird der einspurige Bereich durch eine zweispurige Ausweichstelle unterbrochen. Die Einfahrt in die einspurigen Teilbereiche wird durch insgesamt vier Ampeln A1 bis A4 gesteuert.



Die Ampel A1 kann erst dann auf grün schalten, wenn A2 rot zeigt; umgekehrt kann A2 erst dann auf grün schalten, wenn A1 rot zeigt. Entsprechendes gilt für A3 und A4. Die Ampeln arbeiten in wechselnden zeitlich getakteten Rot- und Grünphasen, zusätzlich unterstützt durch Sensoren. Dadurch wird sichergestellt, dass eine Ampel nur dann auf grün schaltet, wenn der unmittelbar folgende einspurige Bereich völlig frei ist.

Geben Sie an, was man in der Informatik allgemein unter Verklemmung versteht, und führen Sie an einem Beispiel aus, dass die genannten Regeln der Ampelsteuerung nicht ausreichen, um eine Verklemmungssituation in diesem Straßenabschnitt zu verhindern.

(Fortsetzung nächste Seite)

BE

3. Informationen über den Zustand einer Registermaschine werden unter anderem im Statusregister gespeichert. Für jede einzelne dieser Informationen genügt dabei ein Bit, das sogenannte Flag. Die folgende Registermaschine verfügt im Statusregister über ein Negativflag (N) und ein Zeroflag (Z). Das N-Flag bzw. das Z-Flag wird gesetzt, d. h. erhält den Wert 1, wenn das Ergebnis der letzten Operation negativ bzw. Null war, andernfalls wird es zurückgesetzt, d. h. es erhält den Wert 0. Der Befehlssatz der Registermaschine umfasst folgende Befehle:

load x	kopiert den Wert aus Speicherzelle x in den Akkumulator. Dabei werden abhängig vom Wert die Flags angepasst.
store x	kopiert den Wert aus dem Akkumulator in die Speicherzelle x
cmp x	vergleicht den Wert im Akkumulator mit dem Wert in Speicherzelle x. Dazu werden vom Wert im Akkumulator der Wert in Speicherzelle x subtrahiert und die Flags angepasst; der Wert im Akkumulator bleibt dabei unverändert.
jmp x	führt einen unbedingten Sprung zum Befehl in Speicherzelle x aus
jmpz x	springt zum Befehl in Speicherzelle x, falls das Z-Flag gesetzt ist
jmpn x	springt zum Befehl in Speicherzelle x, falls das N- und das Z-Flag nicht gesetzt sind
jmpnp x	springt zum Befehl in Speicherzelle x, falls das N- oder das Z-Flag gesetzt ist
inc	erhöht den Wert im Akkumulator um 1 und passt die Flags an
dec	erniedrigt den Wert im Akkumulator um 1 und passt die Flags an
hold	beendet die Abarbeitung des Programms

(Fortsetzung nächste Seite)

BE

5

- a) Der Zustand der Registermaschine ist vereinfachend festgelegt durch die Werte in Akkumulator (A), Befehlszähler (PC), Statusregister (SR) sowie den beiden Speicherzellen 100 und 101. Im Anfangszustand haben PC und A den Wert 0, das Negativflag (N) und das Zeroflag (Z) im SR sind nicht gesetzt, in Speicherzelle 100 steht der Wert 17 und in 101 der Wert 2.

Gegeben ist folgendes Programm:

```
0:  load 101
2:  jmp 6
4:  hold
6:  dec
8:  store 101
10: load 100
12: inc
14: store 100
16: jmp 0
```

Beschreiben Sie den Zustand der Maschine nach Ausführung des Programms für den gegebenen Anfangszustand und stellen Sie den zugrunde liegenden Algorithmus graphisch dar, z. B. in einem Struktogramm.

5

- b) Das Programm in Teilaufgabe 3a addiert eine nichtnegative ganze Zahl in Speicherzelle 101 zu einer ganzen Zahl in Speicherzelle 100. Modifizieren Sie dieses Programm so, dass beliebige ganze Zahlen addiert werden können.

(Fortsetzung nächste Seite)

BE

4. Jeder Molkereibetrieb in der EU kennzeichnet seine Produkte mit einem Genusstauglichkeitskennzeichen der folgenden Form:

Einer aus mindestens einem und höchstens drei Buchstaben bestehenden Landeskennung folgt ein Leerzeichen oder ein Bindestrich. Optional folgt eine aus höchstens zwei Buchstaben bestehende landesspezifische Kennung, die von einem weiteren Leerzeichen abgeschlossen wird. Dann folgt eine mindestens dreistellige Nummer für den Molkereibetrieb, die aus den Ziffern 0 bis 9 besteht; führende Nullen sind erlaubt. Abgeschlossen wird die Kennung mit einem Leerzeichen, gefolgt von den zwei Buchstaben, die für die landesspezifisch festgelegte Abkürzung der Europäischen Union stehen. Als Buchstaben sind dabei immer nur die Großbuchstaben von A bis Z zulässig. Beispiel für eine Kennung eines deutschen bzw. dänischen Betriebs:
D BY 110 EG bzw. DK–2217 EC.

6 a) Beschreiben Sie die syntaktisch korrekten Genusstauglichkeitskennzeichen in einer einfachen Textnotation (z. B. in EBNF).

Erläutern Sie, warum ein syntaktisch korrektes Genusstauglichkeitskennzeichen nicht unbedingt semantisch korrekt sein muss.

5 b) In Deutschland stehen nach der Landeskennung D oder DE zwei Buchstaben als Bezeichner für das Bundesland, in dem sich der Molkereibetrieb befindet. Entwerfen Sie das Zustandsübergangsdiagramm eines endlichen Automaten, der genau die Genusstauglichkeitskennzeichen der in den Bundesländern Bayern (BY) oder Baden-Württemberg (BW) hergestellten Molkereiprodukte akzeptiert.

7 c) Notieren Sie eine mögliche Implementierung des Automaten aus Teilaufgabe 4b in einer auf dem Deckblatt angegebenen Programmiersprache. Beschränken Sie sich vereinfachend auf die Behandlung eines Zustands, von dem aus mindestens zwei Übergänge möglich sind.

Hinweis: Sie dürfen folgende Methoden einer Klasse ZEICHENKETTE verwenden:

- *laenge()* liefert die Länge der Zeichenkette,
- *zeichenAn(n)* liefert das n -te Zeichen der Zeichenkette; die Zählung beginnt bei 0.

INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

IV.

BE

1. Der Glaskünstler Steve Crystalball stellt im Rahmen einer Gartenschau seine berühmten Crystalball-Stäbe aus. Diese bestehen aus übereinander auf einer Stange aufgereihten schwarzen und weißen Glaskugeln. Das Markenzeichen dieser Stäbe ist, dass sie aus mindestens zwei Glaskugeln bestehen und die Farbe der obersten Kugel bereits weiter unten vorgekommen ist.

2

a) Begründen oder widerlegen Sie, ob es sich bei den nachstehend abgebildeten Stäben um Crystalball-Stäbe handelt.

i)



ii)



Die Menge der bei den Crystalball-Stäben erlaubten Glaskugel-Muster kann als Sprache L über dem Alphabet $\Sigma = \{s, w\}$ interpretiert werden. Dabei steht s für eine schwarze und w für eine weiße Kugel; das erste Zeichen eines Wortes von L entspricht der untersten Kugel.

8

b) Entwerfen Sie ein Zustandsdiagramm für einen endlichen erkennenden Automaten, der die Sprache L akzeptiert.

3

c) Gegeben ist die Grammatik G mit $\Sigma = \{s, w\}$, $V = \{\langle \text{Stab} \rangle, \langle S \rangle, \langle W \rangle, \langle Z \rangle\}$, dem Startsymbol $\langle \text{Stab} \rangle$ und den folgenden Produktionen:

(R1) $\langle \text{Stab} \rangle \rightarrow s\langle S \rangle \mid w\langle W \rangle$

(R2) $\langle S \rangle \rightarrow s\langle S \rangle \mid w\langle Z \rangle \mid s$

(R3) $\langle W \rangle \rightarrow w\langle W \rangle \mid s\langle Z \rangle \mid w$

(R4) $\langle Z \rangle \rightarrow s \mid w$

Zeigen Sie am Beispiel des Wortes „swss“, dass die Sprache L nicht durch die Grammatik G beschrieben wird.

3

d) Ändern Sie die Grammatik G aus Teilaufgabe 1c so ab, dass die Sprache L beschrieben wird.

(Fortsetzung nächste Seite)

BE	
4	<p>e) Geben Sie an, was man in der Informatik allgemein unter einer Verklemmung versteht. Beschreiben Sie zudem ein Szenario im Kontext einer Pflanzaktion bei der Gartenschau, das zu einer Verklemmung führt.</p>
	<p>2. Epidemien sind besonders gefährlich, da eine infizierte Person bereits weitere Personen anstecken kann, bevor sie überhaupt Krankheitssymptome aufweist. Die Zeitspanne zwischen Ansteckung und Auftreten der ersten Symptome heißt Inkubationszeit. Angenommen wird im Folgenden, dass eine neu infizierte Person innerhalb der Inkubationszeit genau zwei, danach jedoch keine weiteren Personen mehr ansteckt. Zu Beginn soll nur eine Person neu infiziert sein. Die rekursive Methode <i>anzahlGeben(n)</i> gibt die Anzahl aller innerhalb der <i>n</i>-ten Inkubationszeit neu infizierten Personen zurück.</p>
3	a) Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache die rekursive Methode <i>anzahlGeben(n)</i> .
2	b) Zeigen Sie anhand der Aufrufsequenz, wie der Rückgabewert des Aufrufs <i>anzahlGeben(3)</i> bestimmt wird. Geben Sie an, wie oft die Methode dabei insgesamt aufgerufen wird.
3	c) Stellen Sie in einer Tabelle die Gesamtanzahl der Aufrufe von <i>anzahlGeben(n)</i> für $n = 0$ bis $n = 4$ zusammen. Schätzen Sie damit den Laufzeitaufwand von <i>anzahlGeben</i> in Abhängigkeit von n ab.

(Fortsetzung nächste Seite)

BE

3. Folgendes Programm einer Registermaschine berechnet die Vorkommalstellen des Logarithmus einer Zahl n zur Basis b (n und b sind positive ganze Zahlen):

```

1:  loadi -1
2:  store 100
3:  load 101
4:  dec
5:  jmpn 13
6:  load 100
7:  inc
8:  store 100
9:  load 101
10: div 102
11: store 101
12: jmp 4
13: hold

```

Dabei bedeuten die im Programm vorkommenden Befehle:

load x	kopiert den Wert aus der Speicherzelle x in den Akkumulator
loadi n	lädt die ganze Zahl n in den Akkumulator
store x	kopiert den Wert aus dem Akkumulator in die Speicherzelle x
inc	erhöht den Wert im Akkumulator um 1
dec	erniedrigt den Wert im Akkumulator um 1
div x	dividiert den Wert im Akkumulator durch den Wert aus der Speicherzelle x (ganzzahlige Division)
jmp x	führt einen unbedingten Sprung zum Befehl in Speicherzelle x aus
jmpn x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator negativ ist
hold	beendet die Abarbeitung des Programms

- 5 a) Notieren Sie alle nacheinander auftretenden Wertekombinationen in den Zellen 100, 101 und 102, wenn zu Beginn der Abarbeitung in Zelle 101 der Wert 25 und in Zelle 102 der Wert 3 abgelegt ist. Geben Sie an, welche Rollen die Speicherzellen 100, 101 und 102 spielen.
- 5 b) Stellen Sie den Algorithmus des Programms graphisch dar, z. B. in einem Struktogramm.

(Fortsetzung nächste Seite)

BE

2

c) Die Ausführung des Befehls `div 102` umfasst folgende Schritte:

1	Operanden holen
2	Befehl decodieren
3	Befehlszähler erhöhen
4	Ergebnis im Akkumulator ablegen
5	Befehl aus dem Speicher holen
6	Operandenadresse bestimmen
7	Befehl ausführen

Ordnen Sie die Schritte geeignet an. Geben Sie dazu die Nummern der Schritte in entsprechender Reihenfolge an.

40