

Abiturprüfung 2011

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>1. Das Speiselokal Gut&Fein rüstet alle Bedienungen mit Handheld-PCs aus, um schnell und unkompliziert Bestellungen und Rechnungen verwalten zu können. Dazu werden alle angebotenen Gerichte mit ihrer Bezeichnung, ihrem Preis, einer eindeutigen ID sowie der Anzahl der aktuell vorhandenen Portionen elektronisch gespeichert. Die Gäste erhalten wie üblich eine gedruckte Speisekarte mit Bezeichnungen und Preisen. Bei der Bedienung eines Tisches wird die Tischnummer, die Anzahl der Gäste an diesem Tisch und eventuell auch der Anlass der Bewirtung aufgenommen. Nennt dann einer der Gäste an diesem Tisch ein Gericht der Speisekarte, so erstellt die Bedienung einen entsprechenden, zunächst offenen Posten für diesen Tisch. Ein Tisch kann beliebig viele Posten bestellen. Jeder Posten ist durch eine eindeutige ID gekennzeichnet. Liefert die Bedienung ein Gericht aus, ändert sie den Status des zugehörigen Postens auf „ausgeliefert“. Soll an einem Tisch bezahlt werden, erstellt sie die Rechnung für alle ausgelieferten Posten und ändert nach dem Bezahlen deren Status auf „bezahlt“. Das Ausstellen von Teilrechnungen ist nicht vorgesehen.</p>
10	<p>a) Erstellen Sie ein Klassendiagramm für obige Situation unter Verwendung der Klassen TISCH, POSTEN und GERICHT. Ergänzen Sie insgesamt zwei geeignete Methoden (jedoch keine Standardmethoden zum Lesen oder Setzen von Attributwerten).</p>
	<p>Getränke können ebenfalls bestellt werden. Dabei soll neben der Bezeichnung, einer ID und dem Preis auch gespeichert werden, wie oft das Getränk noch verkauft werden kann und ob es sich um ein alkoholisches Getränk handelt. Für jede Art von Leistung sollen u. a. die folgenden Methoden existieren:</p> <p><i>istVorhanden()</i>: Diese Methode gibt in einem Wahrheitswert zurück, ob die Leistung tatsächlich geliefert werden kann.</p> <p><i>drucken()</i>: Diese Methode erzeugt eine Bildschirmausgabe mit der Bezeichnung und dem Preis der Leistung, z. B. für den Rechnungsdruck. Alkoholische Getränke sollen dabei mit einem „A“ gekennzeichnet werden.</p>
5	<p>b) Modellieren Sie in einem Klassendiagramm, wie durch das Prinzip der Vererbung Gerichte und Getränke verwaltet werden können.</p>
13	<p>c) Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methoden <i>istVorhanden()</i> und <i>drucken()</i>.</p>

(Fortsetzung nächste Seite)

BE

Die von einem Tisch aufgegebenen Posten werden in einer einfach verketteten Liste BESTELLUNG mit dem Softwaremuster Kompositum unter Berücksichtigung des Konzepts der Trennung von Struktur und Inhalt verwaltet.

- 11 d) Zeichnen Sie ein Objektdiagramm für eine Liste *bestellung*, in der drei Objekte der Klasse POSTEN verwaltet werden. Bei Objekten der Klasse POSTEN darf auf die Angabe von Attributen verzichtet werden.

Stellen Sie außerdem das Ihrem Objektdiagramm zugrunde liegende Klassendiagramm ohne Attribute und Methoden dar.

- 13 e) Die Klasse BESTELLUNG soll eine Methode *rechnungsbetragGeben()* besitzen, die die Summe der Preise aller ausgelieferten Posten zurückgibt. Gehen Sie hierbei davon aus, dass in der Klasse POSTEN zu diesem Zweck bereits folgende Methoden implementiert sind:

<i>istAusgeliefert()</i>	gibt einen Wahrheitswert zurück, der angibt, ob die zugehörige Leistung ausgeliefert wurde oder nicht
<i>preisGeben()</i>	gibt den Preis der zugehörigen Leistung zurück

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der nötigen Methoden.

Das Lokal verwaltet alle Gerichte mit ihrem Preis (in Euro) sowie der vorhandenen Anzahl in einer Datenbank-Tabelle *speisen*:

Bezeichnung	Preis	AnzahlVorhanden
Schnitzel Wiener Art	8,90	35
Schweinebraten	9,50	44
Minestrone	3,90	8

- 3 f) Geben Sie eine Datenbankabfrage (z.B. in SQL) an, mit der Bezeichnung und Preis aller Gerichte, von denen mehr als 10 Portionen vorhanden sind, ausgegeben werden können.

- 4 g) Welche Schritte muss ein objektorientiertes Programm abarbeiten, um die Abfrage der Teilaufgabe 1f auszuführen?

(Fortsetzung nächste Seite)

BE

2. Fabian und Fabienne planen nach dem Abitur eine Reise durch Spanien. Sie wollen auf jeden Fall Madrid, Cordoba, Granada, Sevilla, Barcelona und Toledo besuchen. Mit ihrem „de nada“-Buspass haben sie freie Fahrt auf den in der Karte angegebenen Strecken. Die angegebene Zahl gibt jeweils die Fahrzeit in Stunden an.



6

- a) Die mit diesem Buspass möglichen Verbindungen können durch einen Graphen dargestellt werden. Nennen Sie wesentliche Eigenschaften dieses Graphen und repräsentieren Sie diesen als Adjazenzmatrix.

(Fortsetzung nächste Seite)

BE
11
4
80

b) Eine Klasse GRAPH besitzt folgende Attribute:

- ein Attribut *knotenliste* vom Typ `String[]`, in dem die Städtenamen gespeichert werden;
- ein Attribut *matrix* vom Typ `int[][]`, das die Adjazenzmatrix des Graphen repräsentiert; eine Kante wird durch einen nichtnegativen Eintrag in der Matrix repräsentiert;
- ein Attribut *anzahlKnoten* vom Typ `int`, in dem die Anzahl der Knoten gespeichert ist.

Außerdem besitzt die Klasse GRAPH eine Methode *drucken(knotenNr)*, die den Namen der Stadt, die dem Knoten *knotenNr* entspricht, auf der Konsole ausgibt (*knotenNr* ist dabei der Index des entsprechenden Elements in *knotenliste*). Die Methode *drucken* darf im Folgenden als implementiert vorausgesetzt werden.

Formulieren Sie einen Algorithmus, bei dessen Abarbeitung ausgehend von einem Startknoten alle Knoten eines derartigen „Städteverbindungs-Graphen“ besucht und die Namen der besuchten Städte ausgegeben werden.

Geben Sie insbesondere an, welches zusätzliche Attribut die Klasse GRAPH dazu benötigt.

c) Das Busunternehmen bietet zu einem noch günstigeren Preis einen „super de nada“-Buspass an, der im Unterschied zu obigem Pass alle Verbindungen von und nach Sevilla sowie die Verbindung Barcelona – Granada und umgekehrt nicht enthält.

Begründen Sie, warum der so entstandene Graph (ohne Sevilla) kein Baum ist. Entfernen Sie möglichst wenige Kanten, um aus diesem Graphen einen Baum zu erhalten und zeichnen Sie Ihren Baum mit Granada als Wurzel (ohne Kantenbewertungen).

INF1. MODELLIERUNG UND PROGRAMMIERUNG

II.

BE

1. Das Unternehmen „Fly-High“ hat sich auf Flugbuchungen via Internet spezialisiert. Dazu speichert es von seinen Kunden jeweils Name, E-Mail-Adresse und eine eindeutige Kundennummer.

Von den Flügen stehen folgende Informationen zur Verfügung: Flugdatum, Abflughafen, Zielort, Startzeit und Airline. Darüber hinaus hat jeder Flug eine eindeutige Flugnummer.

Bei der Buchung eines Flugs durch einen Kunden werden das Buchungsdatum, die Anzahl der Plätze, die gewünschte Kategorie (First-, Business- oder Economy-Class) sowie der Gesamtpreis gespeichert und die Buchung mit einer eindeutigen Buchungsnummer versehen.

11

a) Erstellen Sie ein Klassendiagramm für die dargelegte Situation. Ergänzen Sie insgesamt drei geeignete Methoden (jedoch keine Standardmethoden zum Lesen oder Setzen von Attributwerten).

14

b) Notieren Sie in einer der auf dem Deckblatt angegebenen Programmiersprachen für die Klasse KUNDE eine mögliche Implementierung. Bei den Methoden reicht es, wenn Sie sich auf die Methode *buchungHinzufuegen(buchung)* und den Konstruktor beschränken. Berücksichtigen Sie bei der Implementierung folgende Gesichtspunkte:

- Die letzten 20 Buchungen eines Kunden werden in einem Feld der Länge 20 gespeichert, wobei die älteste am Anfang, die jüngste am Ende steht. Beim Hinzufügen einer weiteren Flugbuchung wird gegebenenfalls die älteste Buchung automatisch gelöscht, alle weiteren Buchungen nach vorne verschoben und die neue Buchung ins letzte Feldelement geschrieben.
- Dem Konstruktor werden Anfangswerte für Name, E-Mail-Adresse und Kundennummer übergeben. Ein zusätzliches Attribut, welches die Anzahl der gespeicherten Buchungen des Kunden aufnimmt, wird mit 0 vorbelegt.

(Fortsetzung nächste Seite)

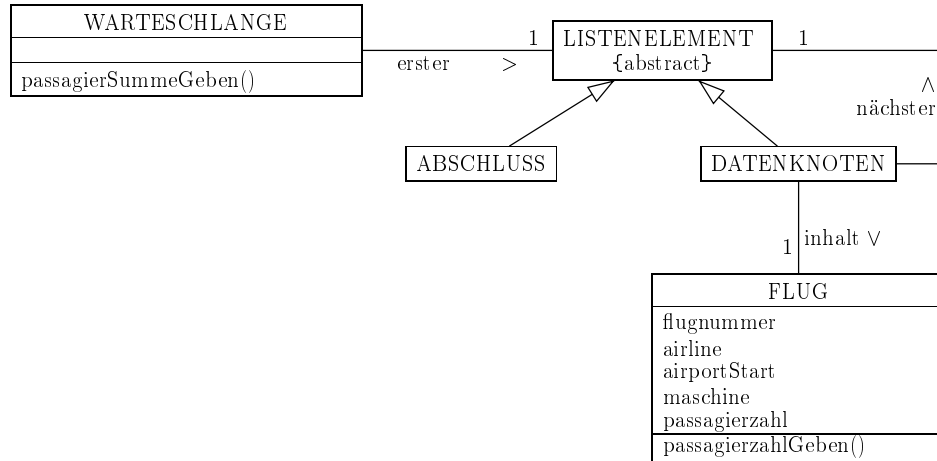
BE	
2	c) Geben Sie zwei Nachteile an, welche die Verwendung eines Feldes für die Verwaltung der Flugbuchungen mit sich bringt.
8	<p>d) Das Unternehmen speichert sämtliche Kundendaten in einem geordneten Binärbaum.</p> <p>Zeichnen Sie einen geordneten Binärbaum mit möglichst wenig Ebenen (d. h. möglichst geringer Höhe) für den Fall, dass dieser nur die Daten von bisher acht Kunden mit den Kundennummern</p> <p style="text-align: center;">4, 7, 10, 28, 33, 50, 61, 74</p> <p>enthält. Verwenden Sie zur Beschriftung der Knoten diese Kundennummern.</p> <p>Wie viele Knoten (einschließlich der Wurzel) müssten im ungünstigsten Fall bei diesem Baum besucht werden, um auf die Daten eines Kunden mit einer bestimmten Kundennummer zugreifen zu können?</p> <p>Geben Sie außerdem an, in welcher Reihenfolge die Knoten Ihres Binärbaums im Falle der Postorder-Traversierung durchlaufen werden.</p>
8	<p>e) Wie viele Knoten hat jeweils ein Binärbaum mit 1, 2, 3 bzw. 4 Ebenen maximal? Stellen Sie anhand dieser Werte eine allgemeine Formel auf, mit welcher die maximale Anzahl der Knoten in Abhängigkeit von der Anzahl k der Ebenen eines Binärbaums berechnet werden kann (Beweis nicht verlangt).</p> <p>Verwenden Sie Ihre Formel um abzuschätzen, wie viele Knoten im ungünstigsten Fall beim Zugriff auf einen bestimmten Kunden besucht werden müssten, wenn die Datenbank insgesamt 10 Millionen Einträge (in Form eines geordneten Binärbaums mit möglichst wenig Ebenen) verwaltet.</p> <p>Erläutern Sie anhand dieser Ergebnisse kurz, warum sich bei einer günstigen Baumstruktur Bäume zur Verwaltung großer Datenmengen besser eignen als verkettete Listen.</p>

(Fortsetzung nächste Seite)

BE

2. Die Software des Towers eines kleinen Flughafens mit einer einzigen Landebahn verwaltet ankommende Flüge in einer Warteschlange mithilfe einer einfach verketteten Liste. Das zugehörige Klassendiagramm sieht im Wesentlichen folgendermaßen aus:

(*Hinweis:* Der Flughafen verwaltet andere Daten über einen Flug als die Firma „Fly-High“ aus Aufgabe 1.)



- 6 a) Zeichnen Sie ein Objektdiagramm für eine Warteschlange mit drei Flugzeugen gemäß obigem Klassendiagramm. Auf die Angabe von Attributen kann bei Objekten der Klasse FLUG verzichtet werden. Geben Sie an, zu welcher Klasse die Objekte jeweils gehören.
- 14 b) Die Methode *passagierSummeGeben()* der Klasse WARTESCHLANGE gibt die Gesamtzahl der Passagiere aller im Anflug befindlichen Flüge zurück, um diese Information an den Zoll weitergeben zu können. Das Attribut *passagierzahl* der Klasse FLUG speichert dabei die Anzahl der Reisenden im betreffenden Flugzeug.
Notieren Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *passagierSummeGeben()* sowie aller Methoden der anderen Klassen der Listenstruktur, die zur Berechnung erforderlich sind. Wenden Sie dabei das Prinzip der Rekursion an.
- 3 c) Welches Entwurfsmuster (Softwaremuster) wird bei der Realisierung der Warteschlange verwendet? Nennen Sie einen Vorteil beim Einsatz von Softwaremustern in der Softwareentwicklung.

(Fortsetzung nächste Seite)

BE

3. Herr Salesman arbeitet in einem Unternehmen mit Niederlassungen in München, Frankfurt, Amsterdam, Oslo und Trondheim. Nachfolgende Tabelle zeigt die Flugmöglichkeiten mit den zugehörigen Flugzeiten; für die Rückflüge gelten jeweils dieselben Flugzeiten.

Von	Nach	Flugzeit
München	Frankfurt	1,5 h
München	Oslo	3 h
München	Amsterdam	2 h
Frankfurt	Amsterdam	1 h
Frankfurt	Oslo	2 h
Frankfurt	Trondheim	3,5 h
Oslo	Trondheim	1 h
Amsterdam	Trondheim	2,5 h

- 8 a) Stellen Sie diese Flugmöglichkeiten als Graph und als Adjazenzmatrix dar. Verwenden Sie bei der Matrix die oben angegebene Reihenfolge (München, Frankfurt, Amsterdam, Oslo, Trondheim).
- 6 b) Herr Salesman soll alle Unternehmensstandorte aufsuchen. Nennen Sie einen Algorithmus zum Graphendurchlauf und skizzieren Sie kurz die Grundidee des Algorithmus. In welcher Reihenfolge besucht Herr Salesman die Orte nach diesem Algorithmus, wenn er in Amsterdam startet und die Knoten in der in Teilaufgabe 3a angegebenen Reihenfolge gespeichert sind?

80

INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

III.

BE

1. Der Befehlssatz einer Registermaschine umfasst die folgenden Befehle:

LOAD x	kopiert den Wert aus Speicherzelle x in den Akkumulator
DLOAD i	lädt die ganze Zahl i in den Akkumulator
STORE x	kopiert den Wert aus dem Akkumulator in die Speicherzelle x
ADD x	addiert den Wert in Speicherzelle x zum Wert im Akkumulator und legt das Ergebnis dort ab
SUB x	subtrahiert den Wert in Speicherzelle x vom Wert im Akkumulator und legt das Ergebnis dort ab
JGE n	springt zum Befehl in Speicherzelle n, falls der Wert im Akkumulator positiv oder Null ist
JLT n	springt zum Befehl in Speicherzelle n, falls der Wert im Akkumulator negativ ist
JUMP n	führt einen unbedingten Sprung zum Befehl in Speicherzelle n aus
END	beendet die Abarbeitung (Ende des Programms)

3

- a) Gegeben ist das Programm:

1: LOAD 102	Der Zustand der Registermaschine wird im Folgenden vereinfachend durch die Inhalte des Akkumulators A und des Befehlszählers BZ sowie der Speicherzellen 101 und 102 beschrieben. Im Anfangszustand ist der Inhalt des Akkumulators 0, der des Befehlszählers 1.
2: SUB 101	
3: JGE 6	
4: LOAD 101	
5: SUB 102	
6: END	

Der Inhalt von Speicherzelle 101 ist 5 und der von Speicherzelle 102 ist 7. Beschreiben Sie tabellarisch die Zustandsänderungen der Registermaschine, während diese das Programm abarbeitet. Geben Sie jeweils auch den Befehl an, der den Zustandsübergang bewirkt.

(Fortsetzung nächste Seite)

BE

3

b) Gegeben ist folgender Algorithmus:

$q = 0$
wiederhole solange $a \geq b$
$a = a - b$
$q = q + 1$
$r = a$

Geben Sie die Werte der Variablen q und r an, die sich nach Abarbeitung des dargestellten Algorithmus mit den Startwerten

i) $a = 15$ und $b = 5$ ii) $a = 13$ und $b = 5$

ergeben. Welche Bedeutung haben allgemein die Werte in q und r nach Ausführung des Algorithmus für positive ganze Zahlen a und b ?

7

c) Schreiben Sie ein Programm für die beschriebene Registermaschine, das den Algorithmus aus Teilaufgabe 1b umsetzt. Gehen Sie davon aus, dass der Wert von a bereits in der Speicherzelle 101 und der von b in der Speicherzelle 102 abgespeichert ist. Die Werte von q bzw. r sollen am Ende in den Speicherzellen 103 bzw. 104 abgespeichert sein. Diese sind zu Beginn mit 0 vorbelegt. Geben Sie zu jedem Befehl auch dessen Speicheradresse an.

8

2. Ein elektronisches Uhrmodul wurde für den Einbau in unterschiedliche Maschinen und Anlagen konstruiert. Die korrekte Eingabe einer Uhrzeit besteht aus der je zweistelligen, durch Doppelpunkt getrennten Angabe von Stunde (im 24-Stunden-Format), Minute und Sekunde. Bei Bedarf kann die Sekundenangabe auf Zehntel- oder sogar Hundertstelsekunden ergänzt werden; als Dezimaltrennzeichen wird das Komma verwendet.

Beschreiben Sie eine solche Zeitangabe in einfacher Textnotation (z. B. in erweiterter Backus-Naur-Form), die nur alltagsübliche Zeitangaben zulässt.

(Fortsetzung nächste Seite)

BE

3. Geldinstitute vergeben Kontonummern nicht willkürlich. Damit Übertragungsfehler mit einer gewissen Wahrscheinlichkeit erkannt werden können, müssen Kontonummern bestimmten Regeln entsprechen, um als gültig akzeptiert zu werden. Bei der INF-Bank ist eine gültige Kontonummer zum Beispiel eine (nichtleere) Folge aus Dezimalziffern mit gerader Anzahl von Ziffern. Sie wird nur dann als gültig akzeptiert, wenn sich jeweils eine Primzahlziffer (aus $\{2; 3; 5; 7\}$) und eine Nichtprimzahlziffer (aus $\{0; 1; 4; 6; 8; 9\}$) abwechseln. Das erste Zeichen darf eine Primzahlziffer oder eine Nichtprimzahlziffer sein. Sie dürfen im Folgenden zur Vereinfachung für eine Primzahlziffer das Zeichen 'P' und für eine Nichtprimzahlziffer das Zeichen 'N' verwenden.

6 a) Geben Sie das Zustandsdiagramm eines endlichen Automaten an, der eine solche Ziffernfolge auf Gültigkeit überprüft.

3 b) Gegeben sind die Ziffernfolgen 4567 und 123456. Überprüfen Sie, welche dieser Ziffernfolgen Ihr Automat akzeptiert, indem Sie jeweils die Folge der durchlaufenen Zustände angeben. Begründen Sie jeweils kurz, warum die Ziffernfolge akzeptiert wird oder nicht.

10 c) Formulieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache für den in Teilaufgabe 3a konstruierten Automaten die Methoden *zustandWechseln* und *folgePruefen*. Dabei soll die Methode *zustandWechseln* ein Zeichen als Parameter übernehmen und dem Attribut *zustand* des Automaten den richtigen Wert zuweisen. Die Methode *folgePruefen* soll eine als Parameter übergebene Zeichenkette auf Gültigkeit als Kontonummer überprüfen. Bei *zustandWechseln* genügt es, wenn Sie beispielhaft die von zwei Zuständen ausgehenden Übergänge implementieren.

Gehen Sie davon aus, dass der Automat ein Attribut *zustand* mit ganzzahligen Werten besitzt, das mit dem Wert 0 vorbesetzt ist. Außerdem dürfen Sie folgende Methoden der Klasse ZEICHENKETTE verwenden:

- *GANZZAHL laenge()*: liefert die Länge der Zeichenkette,
- *ZEICHEN zeichenAnPosition(n)*: liefert das Zeichen an der n-ten Position der Zeichenkette; die Zählung beginnt bei 0.

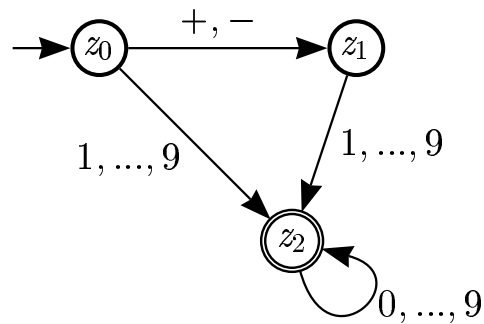
40

INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

IV.

BE
2
4
5

1. Gegeben ist das nachstehende Zustandsdiagramm eines endlichen, erkennenden Automaten.



- 2 a) Geben Sie für jede der folgenden Zahlen an, ob sie vom vorliegenden Automaten akzeptiert wird.

2500 - 13 0 + 3

- 4 b) Beschreiben Sie in einer einfachen Textnotation (z. B. der erweiterten Backus-Naur-Form) die Sprache, die der gegebene Automat akzeptiert.

- 5 c) Eine Dezimalzahl kann als ganze Zahl gefolgt von einem Komma und mindestens einer Ziffer dargestellt werden. Erweitern Sie den bisherigen Automaten so, dass er alle ganzen Zahlen und zudem alle Dezimalzahlen akzeptiert.
Hinweis: Zur Vereinfachung sind auch Zahldarstellungen wie 007,3500 zugelassen.

(Fortsetzung nächste Seite)

BE

2. Eine Registermaschine hat den folgenden Befehlssatz. Dabei ist x die Adresse der Speicherzelle.

`loadi n` lädt die Zahl n in den Akkumulator
`load x` lädt den Wert aus Speicherzelle x in den Akkumulator
`store x` schreibt den Wert aus dem Akkumulator in die Speicherzelle x
`add x` addiert den Wert aus Speicherzelle x zum Wert im Akkumulator und legt diesen dort ab
`sub x` subtrahiert den Wert aus Speicherzelle x vom Wert im Akkumulator und legt ihn dort ab
`inc` vergrößert den Wert im Akkumulator um 1
`dec` verringert den Wert im Akkumulator um 1
`jmp x` springt zum Befehl in Speicherzelle x
`je x` springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
`jg x` springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator positiv ist
`end` beendet die Abarbeitung des Programms

Das folgende Programm zur Multiplikation zweier Zahlen ist in den Speicherzellen 1 bis 10 gespeichert.

```

1    loadi 0
2    store 98
3    load 98
4    add 100
5    store 98
6    load 99
7    dec
8    store 99
9    jg 3
10   end
  
```

- 5 a) In der Speicherzelle mit der Adresse 100 steht zu Beginn die Zahl 6, in der Speicherzelle mit der Adresse 99 steht die Zahl 3. Ermitteln Sie, welche Werte die Speicherzellen 98 bis 100 nach Ablauf des Programms enthalten.
- 4 b) Stellen Sie den in obigem Programm umgesetzten Algorithmus graphisch (z. B. als Struktogramm) dar.

(Fortsetzung nächste Seite)

BE

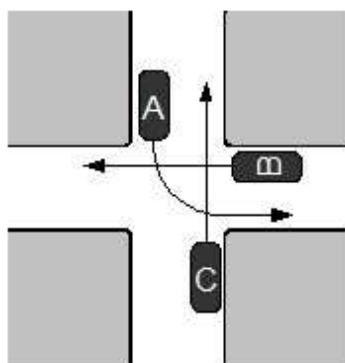
13

- c) Entwickeln Sie für die Registermaschine ein Programm zur Berechnung der Potenz a^n aus positiven ganzen Zahlen a und n . Geben Sie an, in welchen Speicherzellen zu Beginn die Basis a , der Exponent n und am Ende der Potenzwert stehen.

Hinweis: Sie können sich bei der Umsetzung auf obigen Programmcode stützen.

3. Im Stadtstaat Infotasia gelten die gleichen Verkehrsregeln wie bei uns, also z. B. „rechts vor links“ und „Linksabbieger muss Gegenverkehr abwarten“. Die Infotasier halten sich sehr genau an diese Vorfahrtsregeln.

In nachstehender Verkehrskreuzung haben die Fahrzeuge die durch Pfeile angezeigten Zielrichtungen.



5

- a) Was versteht man in der Informatik allgemein unter einer Verklemmung? Erläutern Sie zudem, inwiefern es sich bei der dargestellten Situation um eine Verklemmung handelt.

2

- b) Zur Verhinderung solcher Verklemmungen beschließt Infotasia, künftig an jeder Kreuzung einen Kontrollmechanismus einzuführen. Dieser lässt jeweils nur ein Fahrzeug in den Kreuzungsbereich hineinfahren. Beschreiben Sie knapp ein Konzept der Informatik, welches einem solchen Kontrollmechanismus entspricht.

40