

# Abiturprüfung 2014

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. Viele Firmen sind an anderen Firmen beteiligt. Durch Fusionen, Kooperationen oder Veräußerungen kann es vorkommen, dass Firmen auch gegenseitig unterschiedlich große Firmenanteile voneinander besitzen.</p> <p>Firma A besitzt 49 Prozent der Anteile von Firma B und 12 Prozent von Firma C. Letztere besitzt wiederum 3 Prozent von Firma A. Die Firmen E und F sind jeweils zu 25 Prozent an Firma G beteiligt, die wiederum Anteile in Höhe von 84 Prozent an Firma C hält. Firma F besitzt außerdem 10 Prozent der Firmenanteile von Firma D, umgekehrt hält Firma D einen Anteil von 33 Prozent von Firma F. Die Firmen E und B haben sich im Zuge einer engen Kooperation gegenseitig jeweils 7 Prozent ihrer Firmenanteile überschrieben.</p>
6	<p>a) Stellen Sie die dargelegte Situation mithilfe eines geeigneten Graphen dar. Begründen Sie, dass hier ein ungerichteter Graph ungeeignet ist.</p>
4	<p>b) Geben Sie die zum Graphen von Teilaufgabe 1a gehörige Adjazenzmatrix an.</p>
10	<p>c) Bei der Implementierung wird die Adjazenzmatrix aus Teilaufgabe 1b im zweidimensionalen Feld <i>adjazenzmatrix</i> repräsentiert; das Attribut <i>anzahl-Knoten</i> steht für die Knotenanzahl.</p> <p>Formulieren Sie unter Verwendung dieser Attribute einen Algorithmus zum Graphendurchlauf. Geben Sie anschließend die Reihenfolge der besuchten Knoten an, wenn Sie Ihren Algorithmus auf den Graphen aus Teilaufgabe 1a mit A als Startknoten anwenden.</p> <p>Geben Sie zudem die Bedeutung der Menge der besuchten Knoten im beschriebenen Anwendungsfall an.</p>

(Fortsetzung nächste Seite)

BE

2. Seit dem Schengener Abkommen wird von Kontrollen an den Grenzen zwischen den daran beteiligten Staaten abgesehen. Während an diesen Grenzübergängen meist freie Fahrt herrscht, wird bei der Einreise mit einem KFZ in einen Staat, der nicht dem Schengener Abkommen beigetreten ist, oft eine genaue Kontrolle durchgeführt, bei der u. a. folgende Daten aufgenommen werden: der Staat, in dem das Fahrzeug zugelassen ist, das KFZ-Kennzeichen, die Anzahl der Insassen und das Einreisedatum mit Uhrzeit.

Die Daten sämtlicher in ein bestimmtes Land einfahrenden PKW werden in einem lexikographisch geordneten Binärbaum gespeichert. Dabei dient das Kennzeichen inklusive vorangestellter Länderkennung als Schlüssel. Im Folgenden soll zur Vereinfachung nur der Teilbaum betrachtet werden, der Fahrzeuge mit Starnberger Kennzeichen (D STA) verwaltet.

- 4 a) In einer bestimmten Woche werden folgende PKW mit Starnberger Kennzeichen an der Grenze in angegebener Reihenfolge erfasst: D STA ...

CH 325 RJ 86 SV 131 MV 255 SB 11 AT 74 BU 290 SK 905

Zeichnen Sie den dabei entstehenden Baum.

*Hinweis:* Statt z. B. „D STA CH 325“ können Sie abkürzend „CH 325“ schreiben.

- 3 b) Geben Sie an, in welcher Reihenfolge die Knoten im Falle der Postorder-Traversierung durchlaufen werden.

- 2 c) Der Wagen mit dem Kennzeichen D STA SV 131 verlässt das Land wieder und der betreffende Knoten soll aus dem geordneten Binärbaum gelöscht werden. Beschreiben Sie kurz eine mögliche Idee zur Lösung dieser konkreten Problemstellung.

- 6 d) Beschreiben Sie die Struktur eines geordneten Binärbaums mit 7 Millionen Knoten, der für Suchanfragen besonders gut geeignet ist. Schätzen Sie in diesem Fall die Anzahl der benötigten Ebenen ab.

(Fortsetzung nächste Seite)

BE

3. Für die Zentrale eines kleinen Taxiunternehmens soll eine Software entwickelt werden, die folgende Anforderungen erfüllt:

- Alle angestellten Fahrer werden mit Name und Personalnummer sowie alle Fahrzeuge des Unternehmens mit Kennzeichen verwaltet.
- Jedes Taxi verfügt über ein eigenes GPS-Ortungsgerät, dessen Signal jederzeit in der Zentrale empfangen wird, so dass der aktuelle Aufenthaltsort des Taxis bei Bedarf am Bildschirm in der Zentrale angezeigt werden kann. Bei der Bildschirm-Darstellung wird laufend auf Kartenmaterial aus dem Internet zugegriffen.
- Sämtliche Fahrten werden registriert. Bei Beginn einer neuen Fahrt gibt der Fahrer das Ziel in den Bordcomputer ein, das zusammen mit seiner Personalnummer und dem Kennzeichen des Taxis sowie Datum, Startzeit und -ort an die Zentrale gesendet wird. Der Startort wird dabei vom GPS-System automatisch ermittelt. Bei Beendigung einer Fahrt wird die Ankunftszeit an die Zentrale übermittelt.

16

a) Entwickeln Sie ein Klassendiagramm für die Software, die es dem Taxiunternehmer ermöglicht, oben genannte Vorgänge von seiner Zentrale aus durchzuführen. Das Klassendiagramm muss wenigstens vier sinnvolle Methoden enthalten, jedoch keine Standardmethoden zum Lesen und Setzen von Attributwerten.

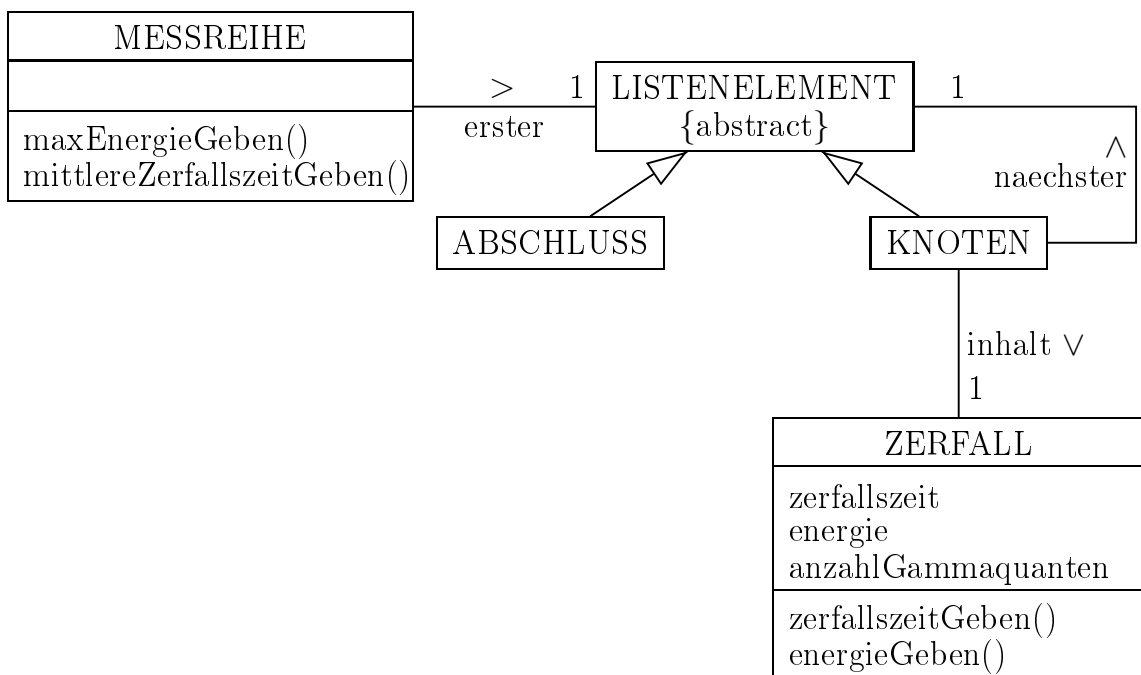
4

b) Geben Sie die grundlegende Idee der Strukturierung von Softwareprojekten in Phasen unter Angabe von mindestens vier Phasen an. Eine Beschreibung der einzelnen Phasen ist nicht verlangt.

(Fortsetzung nächste Seite)

BE

4. Im Jahr 2008 wurde in einem Experiment an der Gesellschaft für Schwerionenforschung in Darmstadt der doppelt-magische Kern  $^{100}\text{Sn}$  erstmals mit ausreichender Statistik produziert. Während des dreiwöchigen Experiments wurden von jedem der über hundert dabei aufgetretenen Zerfallsereignisse folgende Daten registriert: Zerfallszeit (Lebensdauer des Kerns in s), Energie (Zerfallsenergie in J) und Anzahl der emittierten Gammaquanten. Dabei wurden Zerfallszeit und Energie jeweils als Fließkommazahl gespeichert. Sämtliche Zerfalls-Messdaten werden jetzt in einer einfach verketteten Liste mithilfe des Softwaremusters Kompositum gemäß dem abgebildeten Klassendiagramm gespeichert.



5

- a) Zeichnen Sie ein Objektdiagramm für eine Messreihe mit drei Zerfällen gemäß obigem Klassendiagramm. Auf die Angabe von Attributen kann bei den Objekten der Klasse ZERFALL verzichtet werden. Geben Sie an, zu welcher Klasse die Objekte jeweils gehören.

(Fortsetzung nächste Seite)

BE
20
80

- b) Die Methode *mittlereZerfallszeitGeben()* der Klasse MESSREIHE gibt das arithmetische Mittel aller erfassten Zerfallszeiten, die Methode *maxEnergieGeben()* die größte gemessene Energie zurück.
- Notieren Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung aller Klassen. Bezüglich der Methoden können Sie sich auf die beiden oben genannten und alle dazu benötigten Methoden beschränken. Geben Sie aber auch die Konstruktoren der Klassen MESSREIHE, KNOTEN und ZERFALL an. Wenden Sie soweit wie möglich das Prinzip der Rekursion an.

## INF1. MODELLIERUNG UND PROGRAMMIERUNG

### II.

BE

1. Im Internetportal des Buchgroßhändlers „Superbuch“ weisen sich Kunden durch eine Kundennummer und ein Passwort aus. Neukunden können sich unter Angabe von Name, Adresse und Passwort registrieren; ihnen wird nach Überprüfung der Daten auf Vollständigkeit eine Kundennummer zugeteilt. Nach erfolgter Anmeldung steht ein virtueller Warenkorb zur Verfügung, in den Artikel abgelegt werden können. Dabei werden in jedem Warenkorbbeitrag der Artikel und die gewünschte Anzahl gespeichert. Artikel können aus dem Warenkorb wieder entfernt werden; die für einen Artikel gewünschte Anzahl kann bis zum Abschluss der Bestellung jederzeit geändert werden. Wenn der Kunde die Bestellung abgeschlossen hat, wird die Zahlungsart abgefragt. Anschließend muss der Kunde die Bestellung endgültig bestätigen, um den Kauf zu tätigen.

Eine Klasse ARTIKEL mit dem Attribut *artikelNummer* und allen gegebenenfalls notwendigen Methoden sowie eine Klasse KUNDENVERWALTUNG mit der Methode *kundennummerErzeugen* können für alle Teilaufgaben vorausgesetzt werden.

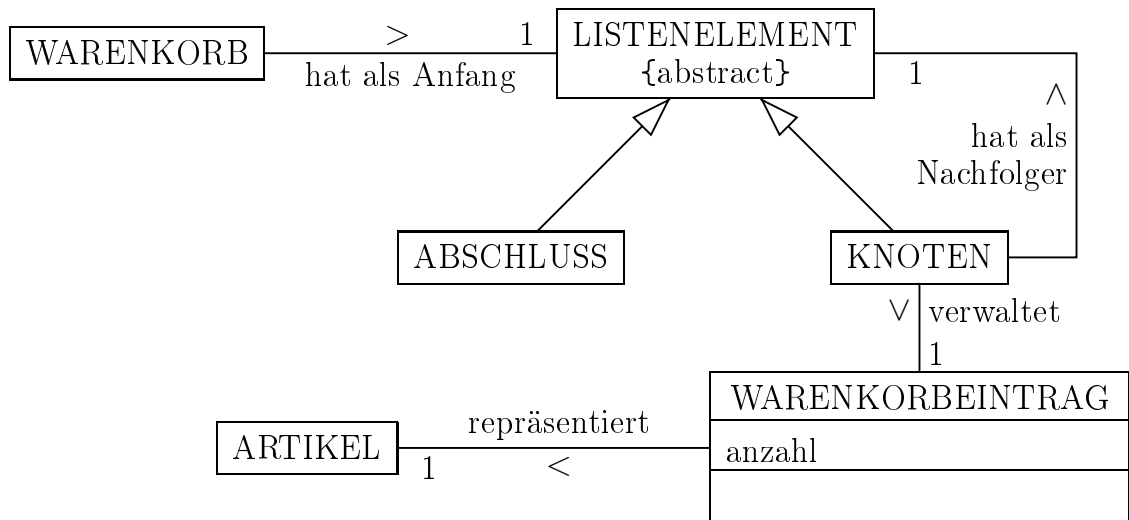
- 5 a) Erstellen Sie ein Sequenzdiagramm, das den beschriebenen Registriervorgang für einen Neukunden wiedergibt.
- 8 b) Geben Sie ein Klassendiagramm an, das die für den oben beschriebenen Bestellvorgang nötigen Klassen darstellt.
- 8 c) Der Bestellvorgang kann durch einen Automaten modelliert werden. Geben Sie ein entsprechendes Zustandsübergangsdiagramm mit aussagekräftigen Bezeichnungen der Zustände und Zustandsübergänge an. Ein eventueller Abbruch der Bestellung durch den Kunden muss nicht berücksichtigt werden.

(Fortsetzung nächste Seite)

BE

12

- d) Die Waren im Warenkorb werden in einer Liste gemäß dem folgenden Klassendiagramm verwaltet:



Notieren Sie unter Verwendung des Prinzips der Rekursion in einer auf dem Deckblatt angegebenen Programmiersprache eine Methode *gesamtbruttopreisGeben()*, welche den Gesamtpreis aller Artikel im Warenkorb zurückgibt. Die Implementierung von *gesamtbruttopreisGeben()* ist für alle relevanten Klassen anzugeben. Eine Methode *bruttopreisGeben()* der Klasse **ARTIKEL** kann als vollständig implementiert vorausgesetzt werden.

Notieren Sie außerdem in allen betroffenen Klassen eine Methode *artikelSuchen(nummer)*, die den Warenkorbeintrag zu dem mit *nummer* bezeichneten Artikel zurückgibt, falls er schon im Warenkorb vorhanden ist; andernfalls wird die leere Referenz zurückgegeben.

2. Superbuch unterscheidet künftig bei seinen Artikeln die Kategorien Bücher, CDs, DVDs und „Weitere Artikel“. Dafür sollen bei allen Artikeln unter anderem der Artikelname und Schlagwörter für die Artikelsuche genannt werden. Bei Büchern sind zusätzlich die Seitenzahl und die Ausführung (Taschenbuch, gebundene Ausgabe, Lederrücken etc.) abrufbar, bei den CDs die Anzahl der Titel und bei den DVDs die Angabe, ob Bonusmaterial vorhanden ist. Im Attribut *preis* aller Artikel ist der Nettopreis gespeichert. Während für alle Bücher der feste Mehrwertsteuersatz 7% und für alle CDs und DVDs der feste Mehrwertsteuersatz 19% gilt, wird bei den weiteren Artikeln auch der jeweils zutreffende Mehrwertsteuersatz gespeichert.

(Fortsetzung nächste Seite)

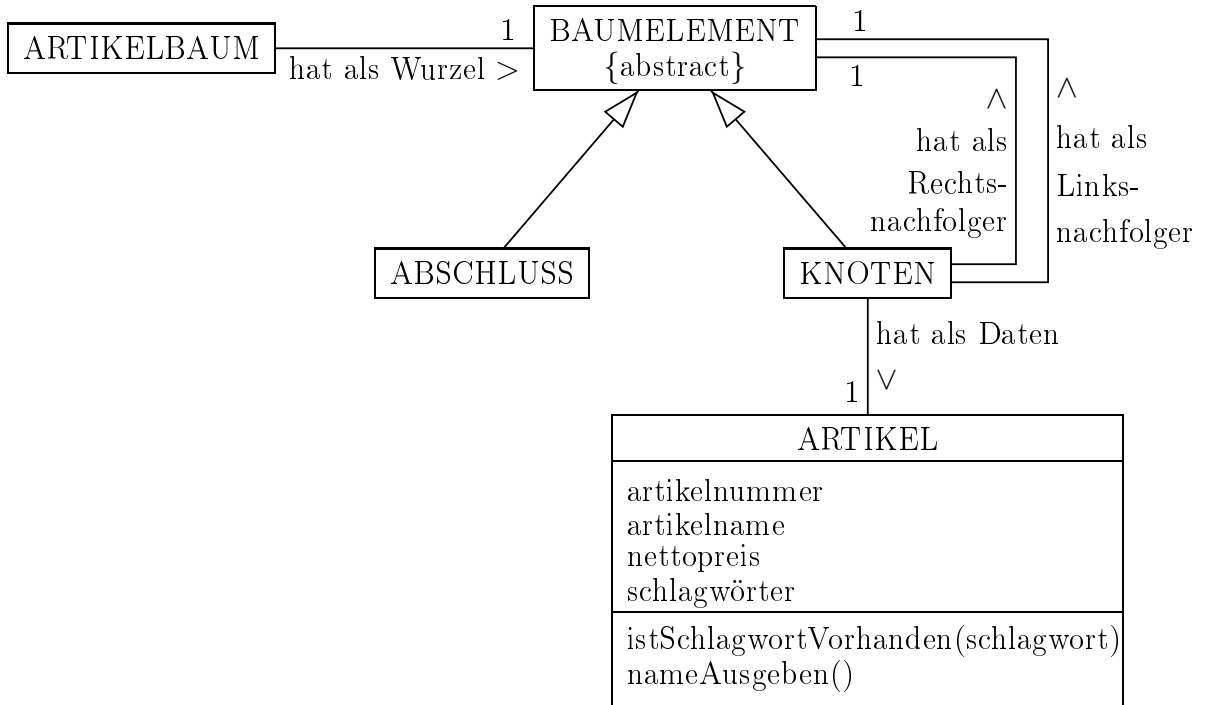


BE	
7	a) Begründen Sie kurz, warum diese Situation sinnvollerweise nach dem Prinzip von Generalisierung und Spezialisierung modelliert wird, und geben Sie ein entsprechendes Klassendiagramm an. Dabei soll neben den genannten Attributen die Methode <i>bruttopreisGeben()</i> aufgeführt werden. Begründen Sie kurz Ihre Modellierungsentscheidung bezüglich der Einbettung dieser Methode in die Klassenstruktur.
5	b) Geben Sie eine Implementierung der Methode <i>bruttopreisGeben()</i> für alle relevanten Klassen aus Teilaufgabe 2a in einer auf dem Deckblatt festgelegten Programmiersprache an.
	3. Um für Großkunden eine Hotlinebetreuung einrichten zu können, soll die Klasse KUNDE als Unterklassen die Klassen KLEINKUNDE und GROSSKUNDE mit entsprechend angepassten Attributen und Methoden erhalten. Zusätzlich soll die Verwaltung der Hotline eingerichtet werden. Die Firma COM14 erhält den Auftrag für die notwendigen Anpassungen und Ergänzungen.
8	a) Nennen und beschreiben Sie kurz die typischen Phasen zur Realisierung eines Softwareprojekts. Erläutern Sie auch den Begriff „Meilenstein“ und geben Sie einen Meilenstein an, der zu der von Ihnen beschriebenen Realisierung des Softwareprojekts passt.
4	b) Die Arbeiten zur Auftrennung der Klasse KUNDE und die Arbeiten an der Hotlineverwaltung können durch parallel arbeitende Entwicklerteams geleistet werden. Die Klasse GESPRACHSVERWALTUNG wird dabei von mehreren Entwicklerteams bearbeitet. Beschreiben Sie anhand der gemeinsamen Nutzung dieser Ressource den Begriff „kritischer Abschnitt“ und erläutern Sie an diesem Beispiel ein Verfahren zur Koordination des Zugriffs auf gemeinsam genutzte Ressourcen.

(Fortsetzung nächste Seite)

BE

4. Wie bei großen Datenmengen üblich, werden u. a. die Artikeldaten in einer Datenbank gespeichert. Aus Effizienzgründen sind die wichtigsten Artikeldaten aber auch zusätzlich im Arbeitsspeicher verfügbar. Dort werden sie gemäß dem folgenden Klassendiagramm in einem geordneten Binärbaum verwaltet; als Schlüssel dient die Artikelnummer.

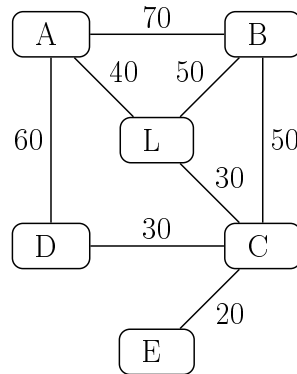


- 8 a) Geben Sie für alle relevanten Klassen in einer auf dem Deckblatt festgelegten Programmiersprache eine mögliche Implementierung der Methode *artikelAuflisten(schlagwort)* an, welche geordnet nach der Artikelnummer die Namen all der Artikel ausgibt, die das gegebene Wort unter ihren Schlagwörtern haben. Die Methoden *istSchlagwortVorhanden(schlagwort)* und *nameAusgeben()* der Klasse **ARTIKEL** stehen dafür bereits zur Verfügung. Erläutern Sie kurz, warum Ihre Implementierung beide Anforderungen an die Ausgabe erfüllt.
- 2 b) Betrachtet der Kunde einen Artikel genauer, müssen weitere Attribute aus der Datenbank gelesen werden. Geben Sie eine Datenbankabfrage (z. B. in SQL) an, die alle Daten des Artikels mit der Artikelnummer 123456 aus der Tabelle *artikel* abruft.

(Fortsetzung nächste Seite)

BE

5. Vom Auslieferungslager L aus werden unter anderem die Orte A, B, C, D und E beliefert. Die Entfernungen in Kilometern gehen aus dem abgebildeten Graphen hervor.



- 3 a) Geben Sie die zu diesem Graphen gehörende Adjazenzmatrix an.
- 4 b) Geben Sie in einer auf dem Deckblatt festgelegten Programmiersprache eine Implementierung der Klasse GRAPH an, die obigen Graph mithilfe der Adjazenzmatrix darstellen kann. Beschränken Sie sich innerhalb der Klasse auf die Attribute und den Konstruktor. Weitere Methoden müssen nicht angegeben werden; die Daten müssen nicht eingetragen werden.
- 6 c) Bei der Auslieferung sollen sämtliche Orte angefahren werden. Beschreiben Sie basierend auf Ihrer Lösung zu Teilaufgabe 5b einen Ihnen bekannten Algorithmus, der, ausgehend von einem gegebenen Knoten, den Graphen durchläuft.

80

## INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

## III.

BE

1. Ein Strichcode besteht aus einer Folge unterschiedlich breiter, abwechselnd schwarzer und weißer Streifen. Jeder Strichcode beginnt und endet mit einem schwarzen Streifen. Auch ein einzelner schwarzer Streifen, der gleichzeitig Beginn und Ende markiert, kann einen gültigen Strichcode darstellen. Schwarze Streifen einfacher Breite werden mit „s“ beschrieben, schwarze Streifen doppelter Breite mit „ss“, schwarze Streifen dreifacher Breite mit „sss“. Analog werden weiße Streifen einfacher, doppelter bzw. dreifacher Breite mit „w“, „ww“ bzw. „www“ beschrieben. Noch breitere Streifen sind nicht vorgesehen. Der nachfolgend dargestellte Strichcode wird nach diesen Vorgaben durch „sswssswswsws“ repräsentiert.



Peter schlägt zur Erzeugung von Strichcodes eine Grammatik mit dem Alphabet  $\{s, w\}$ , der Menge der Nichtterminale  $\{\langle S \rangle, \langle \text{Paar} \rangle, \langle \text{Schwarz} \rangle, \langle \text{Weiß} \rangle\}$ , der Startvariablen  $\langle S \rangle$  und folgenden Produktionsregeln vor:

$$\begin{aligned} \langle S \rangle &\rightarrow \langle \text{Paar} \rangle \langle \text{Schwarz} \rangle \mid \langle \text{Paar} \rangle \langle \text{Paar} \rangle \langle \text{Schwarz} \rangle \mid \\ &\quad \langle \text{Paar} \rangle \langle \text{Paar} \rangle \langle \text{Paar} \rangle \langle \text{Schwarz} \rangle \\ \langle \text{Paar} \rangle &\rightarrow \langle \text{Schwarz} \rangle \langle \text{Weiß} \rangle \\ \langle \text{Schwarz} \rangle &\rightarrow s \mid ss \mid sss \\ \langle \text{Weiß} \rangle &\rightarrow w \mid ww \mid www \end{aligned}$$

*Hinweis:* Der Pfeil bedeutet, dass bei Anwendung der Regel das Nichtterminalsymbol links durch einen der Terme rechts ersetzt wird. Der senkrechte Strich trennt Alternativen.

- 2 a) Weisen Sie z. B. mithilfe einer Ableitung nach, dass der oben dargestellte Strichcode zu der von dieser Grammatik erzeugten Sprache gehört.
- 4 b) Begründen Sie, warum man mit der von Peter angegebenen Grammatik weder beliebig lange Strichcodes noch Codes, die nur aus einem einzigen schwarzen Streifen bestehen, erzeugen kann. Passen Sie obige Produktionsregeln so an, dass beides möglich ist.

(Fortsetzung nächste Seite)

BE	
6	<p>c) Entwerfen Sie ein Zustandsdiagramm für einen endlichen erkennenden Automaten, welcher beliebig lange Strichcodes nach obigen Vorgaben akzeptiert.</p> <p>2. In Kitamrofni werden die Preistafeln an der Biogas-Tankstelle noch per Hand geschrieben. Nachdem die Lieferfirma telefonisch durchgegeben hat, um welchen Betrag der Preis erhöht bzw. erniedrigt wird, geht der Mitarbeiter, der den Telefonanruf entgegengenommen hat, folgendermaßen vor:</p> <p>A) Er geht zur Preistafel und schreibt sich den alten Preis ab.</p> <p>B) Er berechnet den neuen Preis und erstellt eine neue Preistafel.</p> <p>C) Er hängt die neue Preistafel über die bisherige.</p> <p>Gehen Sie im Folgenden davon aus, dass die Lieferfirma die Veränderung des Preises (nicht den neuen Preis!) häufig durchgibt, dass verschiedene gleichberechtigte Mitarbeiter die Anrufe entgegennehmen und bearbeiten und dass die Arbeitsschritte unterschiedlich lange Zeiten in Anspruch nehmen. Außerdem sind die Teilschritte A, B und C jeweils als nicht unterbrechbare Einheit anzusehen.</p>
3	a) Erläutern Sie, inwiefern hier nebenläufige Prozesse auftreten und welche gemeinsam genutzten Betriebsmittel vorkommen.
5	b) Erläutern Sie anhand eines Sequenzdiagramms eine Situation, bei der durch das beschriebene Verfahren eine fehlerhafte Preistafel ausgehängt wird.
4	c) Beschreiben Sie allgemein ein informatisches Konzept, welches den in Teilaufgabe 2b dargestellten Fehler vermeidet. Machen Sie deutlich, wie das Konzept auf das in Teilaufgabe 2b beschriebene Problem angewandt werden müsste.

(Fortsetzung nächste Seite)

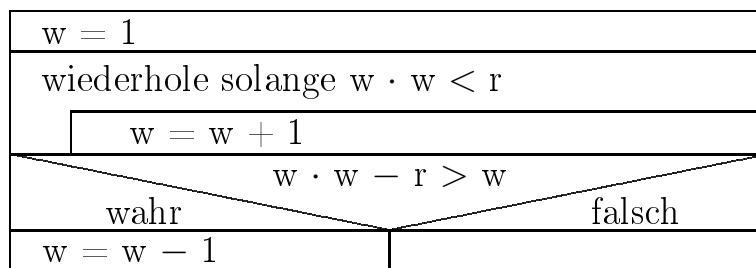
BE

3. Ein gängiges Verschlüsselungsverfahren ist der nach seinen Erfindern Ronald L. Rivest, Adi Shamir und Leonard Adleman benannte RSA-Algorithmus. Seine Sicherheit beruht im Wesentlichen darauf, dass große natürliche Zahlen nur mit erheblichem Aufwand in Primfaktoren zerlegbar sind. Zur Zerlegung einer natürlichen Zahl  $n$ , die das Produkt zweier sechsstelliger Primzahlen ist, benötigt ein PC mit einem heute üblichen Verfahren im Mittel 0,3 s. Ist  $n$  das Produkt zweier siebenstelliger Primzahlen, so benötigt dieser PC im Mittel 3 s, bei achtstelligen sind es bereits 30 s usw.

3 a) Schätzen Sie ab, wie lange ein Superrechner, der 10000-mal so schnell wie dieser PC ist, zur Zerlegung benötigt, wenn  $n$  das Produkt zweier 20-stelliger Primzahlen ist. Geben Sie Ihr Ergebnis in einer sinnvollen Einheit an.

3 b) Erläutern Sie, ob der RSA-Algorithmus unter den oben genannten Gesichtspunkten (Faktorisierungsproblem in zwei Primzahlen als wesentlicher Aspekt und zeitliches Verhalten der Faktorisierung) ein sicheres Verschlüsselungsverfahren ist. Gehen Sie dabei auch auf mögliche Fortschritte in der Rechengeschwindigkeit ein.

4. Das nachstehende Struktogramm beschreibt einen einfachen Algorithmus zur näherungsweise Berechnung der Quadratwurzel einer positiven Zahl  $r$ .



2 a) Geben Sie an, welchen Wert die Variablen  $w$  und  $r$  am Ende haben, wenn  $r$  zu Beginn den Wert 11 hat.

(Fortsetzung nächste Seite)

BE

8

- b) Übertragen Sie den Algorithmus in ein Programm für eine Registermaschine mit nachstehendem Befehlssatz. Machen Sie auch die Speicherzellen deutlich, in denen die Variablen  $r$  und  $w$  zu Beginn und Ende des Programmablaufs stehen.

<code>load x</code>	kopiert den Wert aus der Speicherzelle $x$ in den Akkumulator
<code>loadi n</code>	lädt die ganze Zahl $n$ in den Akkumulator
<code>store x</code>	kopiert den Wert aus dem Akkumulator in die Speicherzelle $x$
<code>add x</code>	addiert den Wert aus der Speicherzelle $x$ zum Wert im Akkumulator
<code>sub x</code>	subtrahiert den Wert aus der Speicherzelle $x$ vom Wert im Akkumulator
<code>addi n</code>	addiert die ganze Zahl $n$ zum Wert im Akkumulator
<code>subi n</code>	subtrahiert die ganze Zahl $n$ vom Wert im Akkumulator
<code>mul x</code>	multipliziert den Wert im Akkumulator mit dem Wert aus der Speicherzelle $x$
<code>div x</code>	dividiert den Wert im Akkumulator durch den Wert aus der Speicherzelle $x$ (ganzzahlige Division)
<code>jmp x</code>	springt zum Befehl in Speicherzelle $x$
<code>jmpn x</code>	springt zum Befehl in Speicherzelle $x$ , falls das Ergebnis der letzten Operation negativ war
<code>jmpz x</code>	springt zum Befehl in Speicherzelle $x$ , falls das Ergebnis der letzten Operation Null war
<code>jmpp x</code>	springt zum Befehl in Speicherzelle $x$ , falls das Ergebnis der letzten Operation positiv war
<code>jmpnn x</code>	springt zum Befehl in Speicherzelle $x$ , falls das Ergebnis der letzten Operation nicht negativ war
<code>jmpnz x</code>	springt zum Befehl in Speicherzelle $x$ , falls das Ergebnis der letzten Operation nicht Null war
<code>jmpnp x</code>	springt zum Befehl in Speicherzelle $x$ , falls das Ergebnis der letzten Operation nicht positiv war
<code>hold</code>	beendet die Abarbeitung des Programms

## INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

## IV.

BE

1. Der Befehlssatz einer Registermaschine umfasst die folgenden Befehle:

load x	kopiert den Wert aus der Speicherzelle x in den Akkumulator
dload n	lädt die ganze Zahl n in den Akkumulator
store x	kopiert den Wert aus dem Akkumulator in die Speicherzelle x
add x	addiert den Wert aus der Speicherzelle x zum Wert im Akkumulator
sub x	subtrahiert den Wert aus der Speicherzelle x vom Wert im Akkumulator
mult x	multipliziert den Wert im Akkumulator mit dem Wert aus der Speicherzelle x
div x	dividiert den Wert im Akkumulator durch den Wert aus der Speicherzelle x (ganzzahlige Division)
jge x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator positiv oder Null ist
jle x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator negativ oder Null ist
jeq x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator Null ist
jne x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator nicht Null ist
jgt x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator positiv ist
jlt x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator negativ ist
jump x	führt einen unbedingten Sprung zum Befehl in Speicherzelle x aus
hold	beendet die Abarbeitung des Programms

(Fortsetzung nächste Seite)



BE

Der Inhalt der Speicherzelle 100 ist der Wert einer positiven ganzen Zahl  $a$ , der Inhalt der Speicherzelle 101 ist der Wert 1, der Inhalt der Speicherzelle 102 ist der Wert 2, der Inhalt der Speicherzelle 103 ist der Wert 3.

4

a) Gegeben ist das folgende Programm:

```

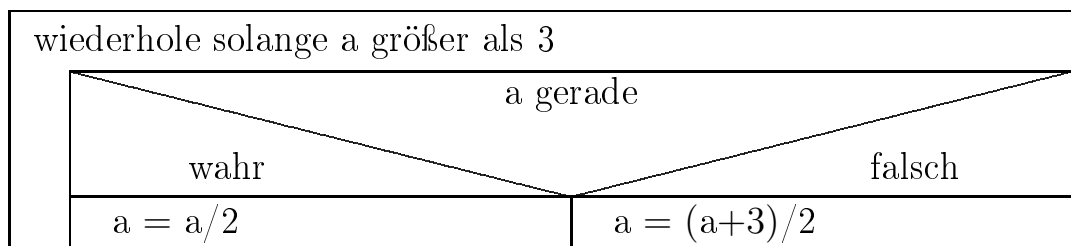
1:  load 100
2:  sub 102
3:  jeq 6
4:  jlt 8
5:  jump 2
6:  store 99
7:  jump 10
8:  load 101
9:  store 99
10: hold

```

Geben Sie in Abhängigkeit von  $a$  den Wert an, der in Speicherzelle 99 abgelegt wird.

9

b) Schreiben Sie ein Programm für die vorgegebene Registermaschine, welches den folgenden Algorithmus umsetzt.



(Fortsetzung nächste Seite)

BE

2. Die folgenden Produktionsregeln gehören zu einer Grammatik, die eine formale Sprache L definiert. Die Regel R1 wird bei der Erzeugung der Wörter stets zuerst angewandt.

R1: Satz  $\rightarrow$  Subjekt Prädikat Satzzeichen

R2: Subjekt  $\rightarrow$  Artikel Substantiv

R3: Prädikat  $\rightarrow$  "lernt" | "schreibt" | "denkt" | "übt" | "feiert"

R4: Artikel  $\rightarrow$  "Der" | "Die"

R5: Substantiv  $\rightarrow$  "Schüler" | "Schülerin" | "Abiturient" | "Abiturientin"

R6: Satzzeichen  $\rightarrow$  "." | ","

*Hinweis:* Der Pfeil bedeutet, dass bei Anwendung der Regel das Nichtterminalsymbol links durch einen der Terme rechts ersetzt wird. Der senkrechte Strich trennt Alternativen.

3 a) Ergänzen Sie die oben stehenden Produktionen zu einer Grammatik der Sprache L.

1 b) Geben Sie ein Wort aus der Sprache L an, das zwar mit den gegebenen Produktionsregeln erzeugt werden kann, das aber dennoch kein korrekter Satz (im Sinne der deutschen Sprache) ist.

3 c) Ändern Sie die Produktionsregeln so ab, dass nur noch die korrekten, vollständigen deutschen Sätze aus der Sprache L erzeugt werden können.

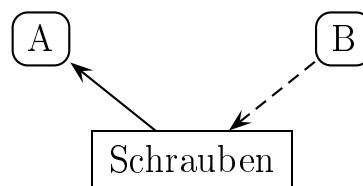
(Fortsetzung nächste Seite)

BE

3. Im Folgenden sollen Prozesse, die auf gleiche Betriebsmittel angewiesen sind, graphisch modelliert werden.

2

a) In der Produktionshalle eines Automobilherstellers kann der Zugriff auf den Schraubenvorratsbehälter immer nur höchstens einem Roboter erlaubt werden. Im Moment sind die Schrauben Roboter A zugeteilt, der ein Armaturenbrett befestigt, so dass der Roboter B, der ebenfalls Schrauben benötigt und daher anfordert, sie gerade nicht bekommen kann. Diese Situation kann wie folgt durch einen Graphen dargestellt werden:



Entscheiden Sie, ob eine Verklemmung vorliegt und begründen Sie Ihre Antwort kurz.

5

b) Vier Köche namens Emil, Fritz, Gerd und Heinz bereiten in der Kantine des Automobilherstellers je ein köstliches Dessert zu. Modellieren Sie die folgende Situation graphisch analog zu Teilaufgabe 3a:

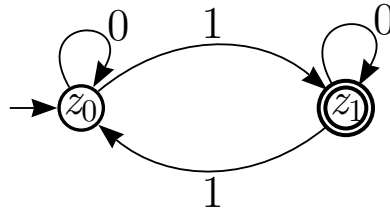
Fritz hält das einzige Sieb der Küche bereit, um Puderzucker auf den Kuchen zu streuen. Heinz hat den einzigen Messbecher, um die notwendige Mehlmenge für eine Torte abzumessen. Während Gerd mit dem gesamten Mehlvorrat auf das Sieb wartet, hat Emil zwar den gesamten Puderzucker, jedoch nicht den Messbecher.

Geben Sie zwei Eigenschaften des entstandenen Graphen an und erläutern Sie, ob aus diesem auf eine Verklemmung geschlossen werden kann.

(Fortsetzung nächste Seite)

BE

4. Zur Kontrolle der Korrektheit bei der Übertragung von Daten verwendet man auf Bitebene häufig eine sogenannte Paritätsprüfung. Dazu ist ein Automat mit folgendem Zustandsübergangsdiagramm gegeben:



*Hinweis:* In diesem Diagramm wird ein Bit durch das Zeichen 0 oder 1 repräsentiert.

- 2 a) Beschreiben Sie die besondere Eigenschaft der Wörter, die dieser Automat erkennt.
- 4 b) Durch diesen Automaten ist eine formale Sprache festgelegt, die auch durch eine Grammatik beschrieben werden kann. Geben Sie deren Produktionen als Syntaxdiagramm an.

Im Folgenden soll nach genau vier Bits entschieden werden, ob die Anzahl der Einsen in diesen vier Bits ungerade ist.

- 1 c) Zeigen Sie an einem Beispiel, dass der oben gegebene Automat zur Lösung dieser Aufgabe ungeeignet ist.
- 6 d) Geben Sie das Zustandsübergangsdiagramm eines endlichen Automaten mit nur einem Endzustand an, der genau die Bitfolgen akzeptiert, für die gilt, dass sie vier Bit lang sind und die Anzahl der Einsen ungerade ist.