Abiturprüfung 2015

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

Das Kartenspiel „Passgenau“ für 2 bis 4 Spieler wird nach folgenden Regeln gespielt:
Jede Karte ist mit einer Farb-Wert-Kombination bedruckt: Es gibt die Farben rot (R), gelb (G), blau (B) und schwarz (S); die Werte reichen von 1 bis 10. Jede mögliche Farb-Wert-Kombination existiert genau einmal.
Zu Beginn des Spiels werden alle Karten gemischt. Jeder Spieler erhält 5 Karten, die er vor den anderen verdeckt in der Hand hält. Eine weitere Karte wird in der Mitte des Spieltisches aufgedeckt; sie bildet den Anfang der gemeinsamen Ablage. Die restlichen Karten werden als sogenannter Talon (verdeckter Kartenvorrat) neben die Ablage gelegt.
Die um den Spieltisch sitzenden Spieler kommen nacheinander im Uhrzeigersinn zum Zug, wobei ein beliebiger Spieler beginnt.
Während seines Zugs darf ein Spieler eine seiner Handkarten auf die Ablage legen, falls sie in Farbe oder Wert mit der obersten Karte der Ablage übereinstimmt. Falls er keine passende Karte besitzt, muss er eine Karte vom Talon ziehen. Passt diese Karte, darf er sie sofort auf die Ablage legen, andernfalls muss er sie zu seinen Handkarten nehmen. Danach ist der nächste Spieler an der Reihe.
Sind die Karten des Talons aufgebraucht, werden alle Karten der Ablage außer der obersten gemischt und als neuer Talon ausgelegt.
Gewonnen hat der Spieler, der zuerst keine Karten mehr auf der Hand hat.

1. Eine Softwarefirma erhält den Auftrag, ein nach obigen Regeln funktionierendes Computerspiel zu entwickeln, und analysiert hierzu zunächst die Struktur des realen Spiels.

10 a) Erstellen Sie ein Klassendiagramm des realen Spiels „Passgenau“ ausschließlich unter Verwendung der Klassen SPIEL, SPIELER, KARTE und KARTENGRUPPE. Modellieren Sie dabei alle auftretenden Gruppen von Karten, beispielsweise den Talon, mithilfe der Klasse KARTENGRUPPE. Auf die Angabe von Methoden kann in dieser Teilaufgabe verzichtet werden.

(Fortsetzung nächste Seite)

BE	
	In Ergänzung zum realen Spiel sollen im Computerspiel neben Personen auch vom Computer simulierte Spieler teilnehmen können.
4	b) Ergänzen Sie das Klassendiagramm aus Teilaufgabe 1a um die Klassen PERSON und PCSPIELER. Passen Sie dabei das ursprüngliche Modell so an, dass zum Ausdruck kommt, dass als Spieler nur Objekte der Klassen PERSON und PCSPIELER existieren können und dass deren Methode *spielzugAusfuehren()* unterschiedlich ablaufen wird.
4	c) Nennen Sie eine Beziehung innerhalb des Klassendiagramms, die Sie bidirektional, also durch gegenseitige Referenzierung der betroffenen Objekte, implementieren würden. Begründen Sie Ihre Entscheidung sachgerecht.
3	d) Die Klasse KARTENGRUPPE kann sowohl unter Verwendung eines Feldes als auch mithilfe einer verketteten Liste implementiert werden. Wägen Sie den Einsatz der beiden Möglichkeiten anhand zweier Aspekte gegeneinander ab.
	Die Softwarefirma entscheidet sich dafür, Kartengruppen als verkettete Listen basierend auf dem Softwaremuster Kompositum sowie unter Berücksichtigung des Konzepts der Trennung von Struktur und Daten zu implementieren. Legen Sie im Folgenden diesen Ansatz zugrunde.
5	e) Zeichnen Sie das Objektdiagramm einer Liste, die nur die Karten S8, R4 und G9 in der angegebenen Reihenfolge enthält. Geben Sie jeweils an, zu welcher Klasse die Objekte gehören.

(Fortsetzung nächste Seite)

BE

Die Klasse KARTENGRUPPE besitzt die im Folgenden beschriebenen Methoden:

- *anzahlGeben()* gibt die Anzahl der in der Liste erfassten Karten zurück.
- *vorneEinfuegen(neueKarte)* fügt die übergebene Karte vorne in der Liste ein.
- *karteGeben(position)* gibt eine Referenz auf die Karte an der angegebenen Listenposition zurück (die Zählung beginnt bei 0).
- *karteEntnehmen(position)* gibt eine Referenz auf die Karte an der angegebenen Listenposition zurück (die Zählung beginnt bei 0) und entfernt die Karte aus der Liste.
- *methodeX()* wird durch folgenden Pseudocode beschrieben:

```
Methode methodeX()
    n ist das Ergebnis von anzahlGeben()
    wiederhole 10 000 mal
        p ist eine ganze Zufallszahl im Bereich von 0 bis einschließlich (n-1)
        vorneEinfuegen(karteEntnehmen(p))
    endeWiederhole
endeMethode
```

4 f) Geben Sie an, was die Methode *methodeX()* leistet und wofür sie in diesem Spiel eingesetzt werden kann.

16 g) Geben Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung aller Klassen der Listenstruktur an. Beschränken Sie sich bei den Methoden auf diejenigen, die zur Implementierung der Methoden *vorneEinfuegen(neueKarte)* und *karteGeben(position)* erforderlich sind und wenden Sie so weit wie möglich das Prinzip der Rekursion an. Gehen Sie davon aus, dass die Klasse KARTE bereits vollständig implementiert ist.

(Fortsetzung nächste Seite)

BE

2. Zur Unterstützung einer möglichen Spielstrategie werden die Handkarten eines Spielers auch als Knoten eines Graphen dargestellt. Eine Kante zwischen zwei Knoten bedeutet, dass die zugehörigen Karten wegen der Übereinstimmung in Farbe oder Wert aufeinander gelegt werden dürfen.

4 a) Ein Spieler hat die Karten S3, G3, B4, G5, B7 und G8 auf der Hand. Zeichnen Sie den zugehörigen Graphen und nennen Sie zwei seiner charakteristischen Eigenschaften.

Eine Strategie in der Klasse PCSPIELER verwendet den beschriebenen Graphen, um eine der Handkarten nach folgenden Kriterien als auszuspielende Karte zu wählen:

- Die Karte darf nach den Spielregeln auf die Ablage gelegt werden;
- von der Karte aus sind möglichst viele andere Handkarten erreichbar;
- sind mehrere Karten in diesem Sinne gleichwertig, wird eine beliebige davon verwendet.

3 b) Der beschriebene Graph wird mithilfe der Adjazenzmatrix *matrix* dargestellt, deren Einträge Wahrheitswerte sind. Beschreiben Sie in Worten, welche Bedeutung der Eintrag *wahr* in der Zelle *matrix[i][j]* hinsichtlich der Spielsituation hat.

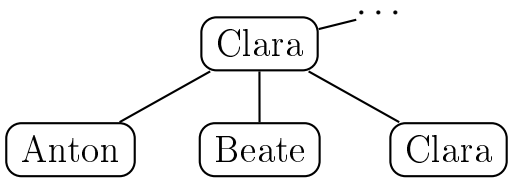
8 c) Um innerhalb des Graphen die von einer Karte aus erreichbaren anderen Karten abzählen zu können, wird ein Algorithmus zum Graphendurchlauf verwendet, der für das Abzählen angepasst werden kann.
Nennen Sie einen Algorithmus zum Graphendurchlauf. Formulieren Sie den für das Abzählen geeigneten Algorithmus; passen Sie hierzu den von Ihnen genannten Algorithmus entsprechend an.

5 d) Nachdem der Gewinner des Spiels feststeht, setzen die restlichen Spieler das Spiel solange fort, bis alle bis auf einen Spieler ihre Handkarten vollständig abgelegt haben. Der letzte Spieler, der noch die Karten S8, S9 und R9 auf der Hand hat, darf noch so viele Karten wie möglich ablegen. Die oberste Karte der Ablage ist S7.
Zeigen Sie an dieser Spielsituation, dass die oben beschriebene Strategie nicht immer den günstigsten Spielverlauf erzeugt.

(Fortsetzung nächste Seite)

BE

3. In einem großen, online ausgetragenen „Passgenau“-Turnier treten immer drei Spieler gegeneinander an. Nur wer ein Spiel gewonnen hat, kommt in die nächste Runde. Der zugehörige Spielplan hat die Struktur eines Baums mit folgenden Eigenschaften: Für jeden bei dem Turnier antretenden Spieler besitzt der Baum ein Blatt. Die übrigen Knoten des Baums beinhalten jeweils den Sieger eines Spiels, in dem die in seinen Kindknoten genannten Spieler gegeneinander angetreten sind.
Beispielsweise zeigt der dargestellte Teilbaum Clara als Siegerin im Spiel gegen Anton und Beate.



In der Wurzel steht schließlich der Turniersieger.
Die Spielerzahl für das Turnier wird so festgelegt, dass sich nach Beendigung des Turniers als Spielplan ein vollständig gefüllter Baum mit n Ebenen ergibt.

3 a) Bestimmen Sie, wie viele Spieler an dem Turnier beteiligt sind, wenn der Baum des Spielplans 3 Ebenen besitzt. Geben Sie auch an, wie groß die Spieleranzahl allgemein bei n Ebenen des Baums ist.

4 b) Bestimmen Sie, wie viele Spiele in dem Turnier stattfinden, wenn der Baum des Spielplans 3 Ebenen besitzt. Geben Sie zudem einen Rechenausdruck für die Anzahl der Spiele innerhalb des Turniers an, wenn der Baum des Spielplans n Ebenen besitzt.

7 c) Zeichnen Sie ein auf dem Entwurfsmuster Kompositum beruhendes Klassendiagramm, welches das Prinzip der Trennung von Struktur und Daten berücksichtigt und den Baum des Spielplans zu jedem Zeitpunkt des Turniers vollständig darstellen kann. Machen Sie dabei deutlich, dass derselbe Spieler an mehreren Positionen des Spielplans stehen kann und dass die Struktur des Plans bereits existiert, bevor die Gewinner der einzelnen Runden ermittelt werden. Auf die Angabe von Attributen und Methoden kann verzichtet werden.

80

Inf1. Modellierung und Programmierung

II.

BE

Eine Softwarefirma ist damit beauftragt, ein Programm zu entwickeln, das Verwaltungsvorgänge einer städtischen Musikschule unterstützen soll. Dazu werden Vorname, Nachname und Geschlecht sowohl der Schülerinnen und Schüler als auch der Lehrkräfte erfasst. Die Identifizierung der Lehrkräfte erfolgt über ein Lehrkraftkürzel, die der Schülerinnen und Schüler durch eine Schülernummer. Von den Musikgruppen werden die Altersstufe, das Niveau, der Unterrichtsort, die Unterrichtszeit und die eindeutige Musikgruppenbezeichnung gespeichert. Jede Musikgruppe wird von genau einer Lehrkraft unterrichtet.

12 1. Die Software soll unter anderem bei der Einrichtung der Musikgruppen eingesetzt werden. Es soll möglich sein, die Schülerliste einer Musikgruppe sowie die Liste aller Lehrkräfte einer Schülerin oder eines Schülers zu drucken. Eine Schülerin oder ein Schüler soll zu einer Musikgruppe hinzugefügt oder aus einer Musikgruppe entfernt werden können.
Modellieren Sie die beschriebene Situation in einem Klassendiagramm. Nutzen Sie dabei das Konzept der Generalisierung. Führen Sie im Klassendiagramm insgesamt vier Methoden an, die keine Standardmethoden zum Lesen oder Setzen eines Attributwerts sind.

2. Weiterhin soll die Software Termine, wie öffentliche Aufführungen und Sonderproben, verwalten. Dazu ist ein Kalender vorgesehen, in den solche Termine eingetragen werden können. Abgelaufene Termine werden automatisch aus dem Kalender entfernt.

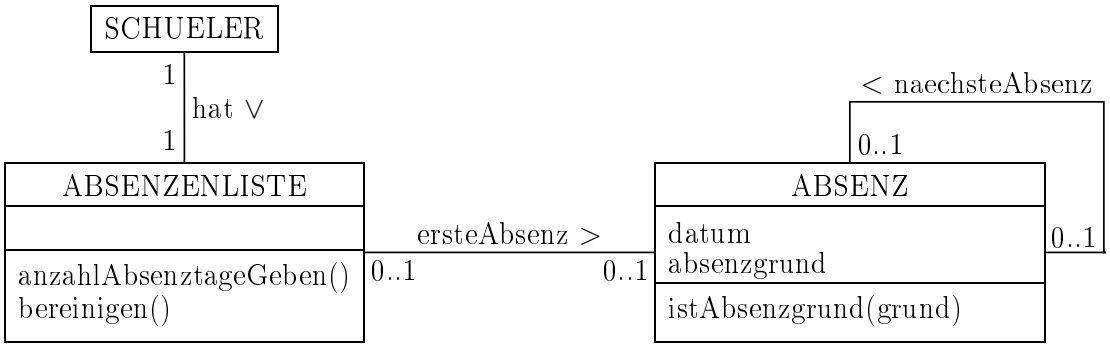
4 a) Ein Mitarbeiter der Softwarefirma soll festlegen, ob die Termine in einem Feld oder in einer einfach verketteten Liste gespeichert werden. Erläutern Sie kurz zwei Aspekte, die im vorliegenden Anwendungsfall für seine Entscheidung relevant sein können.

5 b) Stellen Sie die grundlegende Idee der Datenstruktur Warteschlange dar und beurteilen Sie, ob sie sich zur Verwaltung der Termine eignet.

(Fortsetzung nächste Seite)

BE

3. Schülerinnen und Schüler werden von der Teilnahme an öffentlichen Aufführungen ausgeschlossen, wenn sie im Unterricht zu häufig fehlen. Daher soll das Programm unter Verwendung von einfach verketteten Listen die Schülerabsenzen verwalten. Ein Mitarbeiter der Softwarefirma modelliert die Liste folgendermaßen:



Dabei gilt für die im Klassendiagramm aufgeführten Methoden:

- *istAbsenzgrund(grund)* gibt genau dann den Wahrheitswert *wahr* zurück, wenn der Parameter *grund* (mit dem Datentyp ZEICHENKETTE) den gleichen Wert hat wie das Attribut *absenzgrund*.
- *anzahlAbsenztageGeben()* gibt die Gesamtanzahl aller in der Liste gespeicherten Absenztage zurück.
- *bereinigen()* entfernt alle Absenzen aus der Liste, bei denen als Absenzgrund der Wert „öffentliche Aufführung“ eingetragen ist.

Hinweis: Der Einfachheit halber wird angenommen, dass sich Schülerinnen und Schüler tageweise abmelden und dass jede Absenz eine Abwesenheit von genau einem Tag darstellt.

10

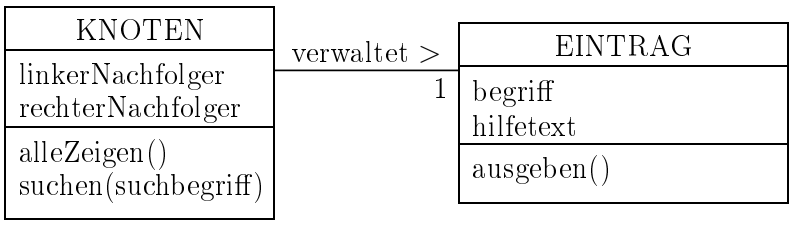
a) Die Projektleiterin akzeptiert den Entwurf zwar im Kern, fordert aber die Verwendung des Entwurfsmusters Kompositum und die Berücksichtigung des Prinzips der Trennung von Struktur und Daten in der hier modellierten Listenstruktur.
Ändern Sie den obigen Entwurf so ab, dass das Klassendiagramm den Vorgaben der Projektleiterin entspricht.
Geben Sie zudem zwei Gründe an, warum die Forderungen der Projektleiterin sinnvoll sind.

(Fortsetzung nächste Seite)

BE

17 b) Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung aller Klassen der Listenstruktur mit den Methoden *anzahlAbsenztageGeben* und *bereinigen*. Beschränken Sie sich bei den Methoden auf diejenigen, die zur Implementierung der Methoden *anzahlAbsenztageGeben* und *bereinigen* erforderlich sind. Legen Sie dabei die verbesserte Listenstruktur aus Teilaufgabe 3a zugrunde und wenden Sie soweit wie möglich das Prinzip der Rekursion an. Gehen Sie davon aus, dass die Klasse ABSENZ bereits vollständig implementiert ist.

4. Die Hilfefunktion des Programms soll zu wichtigen Begriffen Hilfetexte zur Verfügung stellen. Der Benutzer soll entweder nach einem Begriff suchen oder sich alle Begriffe zusammen mit ihren Hilfetexten anzeigen lassen können.
Jeder Begriff wird zusammen mit dem Hilfetext als Eintrag in einem nach Begriffen alphabetisch geordneten Binärbaum verwaltet. Dieser Baum ist nach dem Entwurfsmuster Kompositum implementiert. Einen Ausschnitt des Klassendiagramms zeigt die folgende Abbildung:



Die Methode *ausgeben* der Klasse EINTRAG erzeugt eine Ausgabe des Begriffs zusammen mit seinem Hilfetext, die Methode *alleZeigen* der Klasse KNOTEN dient zur alphabetisch sortierten Ausgabe der Begriffe zusammen mit ihren Hilfetexten.

7 a) In den zunächst leeren Baum werden nacheinander die Begriffe „Zertifikat“, „Abmeldung“, „Anmeldung“ und „Zahlung“ eingefügt. Zeichnen Sie den dadurch entstehenden Baum.
Nun werden viele weitere Elemente eingefügt. Diskutieren Sie, ob der Vorteil des geordneten Binärbaums gegenüber einer Liste beim Suchen nach einem Begriff zum Tragen kommt.

4 b) Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung der Methode *alleZeigen* der Klasse KNOTEN.

(Fortsetzung nächste Seite)

BE

5. Für statistische Zwecke wird die Einteilung aller Musikgruppen in einem Graphen repräsentiert, dessen Knoten die Musikgruppen darstellen. Werden zwei Musikgruppen teilweise von denselben Schülerinnen bzw. Schülern besucht, so besteht zwischen den entsprechenden Knoten eine Kante, deren Gewicht die Anzahl der Schülerinnen und Schüler angibt, die beide Musikgruppen besuchen.

8 a) Insgesamt 48 Kinder haben sich sowohl zur musikalischen Grundbildung als auch zum Kinderchor angemeldet. Ausschließlich für diese Kinder werden zwei gleich große Gruppen zur musikalischen Grundbildung (*mgb1* bzw. *mgb2*) und zwei gleich große Kinderchorgruppen (*kchor1* und *kchor2*) eingerichtet. 7 Schülerinnen und Schüler aus der Musikgruppe *mgb1* und 17 aus der Musikgruppe *mgb2* besuchen den Kinderchor *kchor1*, alle anderen Kinder aus den Gruppen *mgb1* und *mgb2* besuchen den Kinderchor *kchor2*. 10 Kinder des Chors *kchor2* gehen zudem noch in den Flötenunterricht (Musikgruppe *floete1*), aber kein Kind aus *kchor1*.
Zeichnen Sie für die beschriebene Situation einen möglichen Graphen und stellen Sie diesen auch als Adjazenzmatrix dar.

2 b) Nennen Sie zwei Eigenschaften des Graphen aus Teilaufgabe 5a.

3 c) Unter einem zusammenhängenden Graphen versteht man hier einen Graphen, bei dem von jedem Knoten ein Pfad zu jedem anderen Knoten existiert.
Stellen Sie kurz dar, was man aus einem nicht zusammenhängenden Graphen bzgl. der Verteilung der Schülerinnen und Schüler auf die Musikgruppen schließen kann.

8 d) Eine Klasse GRAPH speichert die Knoten in einem Feld *knoten* der Länge n (wobei n für die Gesamtanzahl aller Kurse steht) und die Adjazenzmatrix in einem zweidimensionalen Feld *matrix* der Größe $n \times n$. Existiert zwischen zwei Knoten keine Kante, so wird dies in der Adjazenzmatrix durch den Wert 0 repräsentiert.

(Fortsetzung nächste Seite)

BE

Die Klasse GRAPH hat die folgende Methode *m1* (der Parameter *indizes* dieser Methode ist ein Feld von Knotenindizes, durch die eine Menge von Kursen repräsentiert wird):

```
Methode m1(GANZZAHL[] indizes)
    GANZZAHL m
    setze m auf die Länge des Feldes indizes
    WAHRHEITSWERT b
    zähle i von 0 bis n-1
        setze b auf wahr
        zähle j von 0 bis m-1
            wenn (i=indizes[j]) oder (matrix[i][indizes[j]] > 0)
                setze b auf falsch
            endeWenn
        endeZähle
        wenn b wahr ist
            gib Bezeichnung von knoten[i] aus
        endeWenn
    endeZähle
endeMethode
```

Erläutern Sie unter Angabe der relevanten Schritte, zu welcher Ausgabe die Methode *m1* mit den folgenden Beispieldaten ($n = 5$) führt. Beschreiben Sie zudem, welche Ausgabe die Methode *m1* allgemein erzeugt.

k	0	1	2	3	4
Bezeichnung von knoten[k]	piano1	piano2	jchor1	jchor2	oboe

j	0	1
indizes[j]	0	3

matrix	0	1	2	3	4
0	0	0	18	0	3
1	0	0	0	0	17
2	18	0	0	0	8
3	0	0	0	0	12
4	3	17	8	12	0

Inf2. Theoretische und Technische Informatik

III.

BE

Eine Autohauskette betreibt eine große Zahl von Filialen.

1. Die Geschäftsführer der einzelnen Filialen werden nach folgendem Schema am wöchentlichen Umsatz beteiligt:

Umsatz U	Provision P
bis einschließlich 200.000 Euro	0,2 % vom Umsatz
über 200.000 Euro bis einschließlich 400.000 Euro	0,3 % vom Umsatz
über 400.000 Euro	0,4 % vom Umsatz + 1.000 Euro Bonus

3 a) Formulieren Sie einen Algorithmus, der die Berechnung der Provision gemäß dem genannten Schema umsetzt.

(Fortsetzung nächste Seite)

BE
7

b) Gegeben ist eine Registermaschine mit folgendem Befehlssatz:

`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
`add x`	addiert den Wert aus der Speicherzelle x zum Wert im Akkumulator
`addi n`	addiert die ganze Zahl n zum Wert im Akkumulator
`sub x`	subtrahiert den Wert aus der Speicherzelle x vom Wert im Akkumulator
`subi n`	subtrahiert die ganze Zahl n vom Wert im Akkumulator
`mul x`	multipliziert den Wert im Akkumulator mit dem Wert aus der Speicherzelle x
`muli n`	multipliziert den Wert im Akkumulator mit der ganzen Zahl n
`div x`	dividiert den Wert im Akkumulator durch den Wert aus der Speicherzelle x (ganzzahlige Division)
`divi n`	dividiert den Wert im Akkumulator durch die ganze Zahl n (ganzzahlige Division)
`jmp x`	springt zum Befehl in Speicherzelle x
`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
`hold`	beendet die Abarbeitung des Programms

Schreiben Sie ein Programm für die angegebene Registermaschine, das den Algorithmus von Teilaufgabe 1a umsetzt. Dabei ist Speicherzelle 101 mit dem Umsatz vorbelegt. Die berechnete Provision soll in Speicherzelle 102 abgelegt werden.
Hinweis: Rundungsfehler durch Ganzzahldivision dürfen vernachlässigt werden.

(Fortsetzung nächste Seite)

BE

2. Um in die zentrale Personalabteilung der Autohauskette zu gelangen, muss man vor der Sicherheitstür ein aus genau drei Zeichen bestehendes Passwort eingeben. Dieses wird jährlich gemäß den folgenden Vorgaben der Firma festgelegt:

   - Das Passwort muss mindestens einen Kleinbuchstaben und eine Ziffer enthalten;
   - es darf nicht mit einem der 32 Sonderzeichen (z. B. $*$, §, ...) beginnen;
   - Großbuchstaben sind an keiner Stelle zugelassen.

   Hinweis: Sie dürfen im Folgenden zur Vereinfachung für Kleinbuchstaben das Zeichen b, für Ziffern das Zeichen Z und für Sonderzeichen das Zeichen S verwenden.

5 a) Geben Sie das Zustandsdiagramm eines endlichen, erkennenden Automaten an, der überprüft, ob eine Zeichenkette den Passwortvorgaben der Firma entspricht.

3 b) Beschreiben Sie in einer einfachen Textnotation (z. B. in EBNF) die Syntax der Sprache, die der Automat aus Teilaufgabe 2a akzeptiert.

8 c) Formulieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung für den in Teilaufgabe 2a erstellten Automaten. Dabei soll es u. a. eine Methode *passwortTesten(eingabe)* geben, die überprüft, ob die übergebene Zeichenkette *eingabe* den Vorgaben der Firma entspricht, und die einen entsprechenden Wahrheitswert zurückgibt.
   Es genügt, wenn Sie beispielhaft die von zwei Zuständen ausgehenden Übergänge implementieren.
   Hinweis: Sie dürfen dabei folgende Methoden der Klasse ZEICHENKETTE verwenden:

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

(Fortsetzung nächste Seite)

BE	
	Die Vorgabe der Autohauskette liefert keine besonders sicheren Passwörter, insbesondere wenn die Vorgaben für das Passwort öffentlich bekannt sind.
2	d) Ein Mitarbeiter schlägt vor, die Festlegung, dass mindestens ein Kleinbuchstabe im Passwort vorkommen muss, fallen zu lassen, um die Sicherheit des Passworts zu erhöhen. Erläutern Sie kurz, warum dieser Verbesserungsvorschlag sinnvoll ist.
5	e) Geben Sie zwei weitere Verbesserungsvorschläge für die Vorgabe zur Passwortauswahl an. Prüfen Sie zudem mithilfe einer geeigneten Berechnung, ob ein nach Ihren verbesserten Vorgaben gebildetes Passwort einer Brute-Force-Attacke mit 2 Milliarden Passworteingaben pro Sekunde gut standhalten kann.
	3. Ein zentraler Datenbankserver der Autohauskette verwaltet alle zum Verkauf stehenden Gebrauchtwagen. Am Montagmorgen um 10 Uhr sucht sich Herr Flink im Verkaufsgespräch in der Filiale Bad Kissingen den letzten verfügbaren P300XT eco aus. Zeitgleich möchte Frau Hurtig in der Filiale Niederalteich dasselbe Auto erwerben.
5	a) Erläutern Sie im Sachzusammenhang an einem Sequenzdiagramm die Begriffe Nebenläufigkeit und kritischer Abschnitt.
2	b) Nennen Sie ein Verfahren zur Absicherung des kritischen Abschnitts und geben Sie die wesentliche Idee dieses Verfahrens an.
40	

Inf2. Theoretische und Technische Informatik

IV.

BE

1. Zur Definition aller positiven ganzen Zahlen kann eine Grammatik mit <zahl> als Startsymbol und den folgenden Produktionsregeln verwendet werden:

   <zahl> → <ziffer1> | <zahl> <ziffer>
   <ziffer1> → 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
   <ziffer> → 0 | <ziffer1>

   Dabei bedeutet der Pfeil, dass die syntaktische Variable links durch eine der durch senkrechte Striche getrennten Alternativen auf der rechten Seite ersetzt werden kann.

2 a) Geben Sie eine Ableitung des Wortes „203“ an.

3 b) Zeichnen Sie das Zustandsübergangsdiagramm eines endlichen erkennenden Automaten, der genau die durch die Grammatik beschriebenen Zeichenketten akzeptiert.

Eine einfache maschinennahe Programmiersprache stellt nur die folgenden Befehle zur Verfügung:

```
load  store  add  sub  jmp  jeq  hold
```

Ein Programm besteht aus mindestens einer Programmzeile. Jede Programmzeile beginnt mit einer positiven ganzen Zahl als Zeilennummer; danach folgt ein Befehl. Allen Befehlen außer `hold` folgt eine weitere positive ganze Zahl als Zieladresse. Jede Programmzeile wird durch einen Strichpunkt abgeschlossen.

7 c) Geben Sie eine Grammatik an, mit der alle derartigen Programme beschrieben werden können. Die obigen Produktionsregeln dürfen verwendet werden.

(Fortsetzung nächste Seite)

BE

2. Gegeben sind zwei Algorithmen zur Berechnung der Potenz a^n, wobei n eine nicht-negative ganze Zahl und a eine beliebige positive Zahl ist.

Algorithmus 1:

```
Methode potenz1(ZAHL a, GANZZAHL n)
    wenn n den Wert 0 hat, dann gib 1 zurück
    sonst gib das Produkt aus a und potenz1(a, n-1) zurück
    endeWenn
endeMethode
```

Algorithmus 2:

```
Methode potenz2(ZAHL a, GANZZAHL n)
    wenn n den Wert 0 hat, dann gib 1 zurück
    sonst
        wenn n gerade ist, dann gib das Quadrat
                                von potenz2(a, n/2) zurück
        sonst gib das Produkt aus a und dem Quadrat
                                von potenz2(a, (n-1)/2) zurück
        endeWenn
    endeWenn
endeMethode
```

4 a) Ermitteln Sie für beide Algorithmen die Anzahl der Methodenaufrufe von *potenz1* bzw. *potenz2*, die jeweils zur Berechnung von 5^4 und 5^8 erforderlich sind.

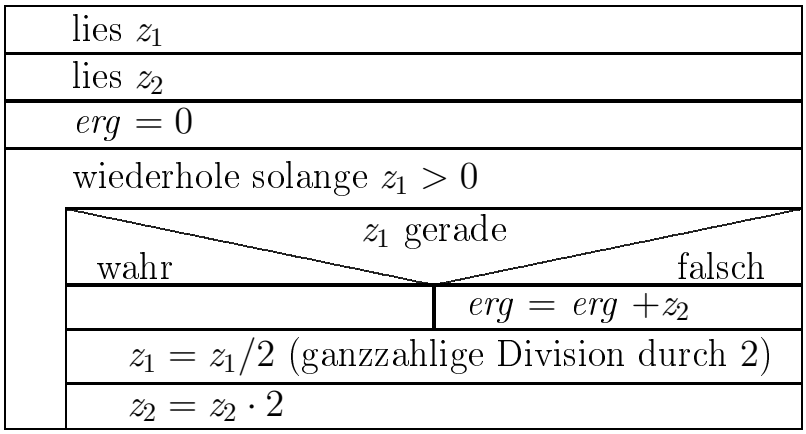
3 b) Geben Sie für beide Algorithmen allgemein an, wie sich die Anzahl der Methodenaufrufe verändert, wenn der Exponent n verdoppelt wird.

4 c) Schätzen Sie für beide Algorithmen ab, wie lange die Berechnung von $5^{1.000.000}$ dauert, wenn pro Methodenaufruf von *potenz1* bzw. *potenz2* eine Dauer von 10^{-6} s veranschlagt wird und sonst keine weiteren Zeiten berücksichtigt werden müssen.

(Fortsetzung nächste Seite)

BE

3. Auf dem ägyptischen *Papyrus Rhind*, der etwa auf das Jahr 1550 v. Chr. datiert wird, ist eine Möglichkeit zur Multiplikation zweier natürlicher Zahlen z_1 und z_2 beschrieben.
Als Struktogramm lässt sich dieser Algorithmus folgendermaßen darstellen:



Das berechnete Produkt steht nach Abarbeitung des Algorithmus in der Variablen *erg*.

4 a) Berechnen Sie mithilfe der beschriebenen ägyptischen Multiplikation schrittweise das Produkt aus $z_1 = 13$ und $z_2 = 5$.

3 b) Nennen Sie die wesentliche Idee des Speichermodells eines Rechners, der nach dem von-Neumann-Prinzip aufgebaut ist. Geben Sie einen Vor- und einen Nachteil dieses Speichermodells an.

Eine Registermaschine besitzt den folgenden Befehlssatz:

`load x`	kopiert den Wert aus der Speicherzelle x 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
`shl`	multipliziert den Wert im Akkumulator mit 2 (shift left)
`shr`	dividiert den Wert im Akkumulator ganzzahlig durch 2 (shift right)
`jump x`	springt zum Befehl in Speicherzelle x
`jeq x`	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator Null ist
`end`	beendet die Abarbeitung des Programms

(Fortsetzung nächste Seite)

BE
3
7
40

c) Bestätigen Sie anhand zweier Beispiele, dass mithilfe des folgenden Programmausschnitts entschieden werden kann, ob Speicherzelle 101 eine gerade oder eine ungerade Zahl enthält.

```
load 101
shr
shl
sub 101
```

d) Schreiben Sie ein Programm für die angegebene Registermaschine, das den Algorithmus des *Papyrus Rhind* umsetzt. Gehen Sie dabei davon aus, dass die beiden positiven ganzzahligen Faktoren z_1 und z_2 bereits in den Speicherzellen 101 und 102 stehen und dass alle weiteren nicht vom Programm belegten Speicherzellen mit dem Wert 0 vorbelegt sind.