

Abiturprüfung 2012

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

Bei den Olympischen Spielen treten Mannschaften und Athleten aus den verschiedensten Ländern der Welt in unterschiedlichen Sportarten gegeneinander an. Bei den Sommerspielen 2012 in London werden Athleten aus ca. 190 Nationen für insgesamt 302 Wettbewerbe in 26 Sportarten erwartet.

1. Im Rahmen der Eröffnungsfeier marschieren die Mannschaften aller teilnehmenden Nationen ins Stadion ein. Jeweils ein Sportler aus der Mannschaft einer Nation trägt dabei die Flagge seines Landes.

Zunächst soll vereinfachend angenommen werden, dass die Nationen in alphabetischer Reihenfolge einziehen.

Die eingesetzte Software verwaltet alle teilnehmenden Nationen mithilfe einer sortierten Liste *nationenliste2012* der Klasse SORTIERTELISTE. In dieser werden die Nationen automatisch in der Reihenfolge des Einzugs angeordnet. Für jede Nation werden der Ländername, die Anzahl der teilnehmenden Athleten und der Name des Flaggenträgers gespeichert. Als Datenstruktur wird eine einfach verkettete, sortierte Liste mit dem Softwaremuster Kompositum unter Berücksichtigung des Konzepts der Trennung von Struktur und Inhalt verwendet.

6 a) Erstellen Sie ein Objektdiagramm, das die Struktur der sortierten Liste mit den vier Nationen Frankreich, Polen, Deutschland und China entsprechend der oben beschriebenen Datenstruktur repräsentiert. Auf die Angabe von Attributen kann bei Objekten der Klasse NATION verzichtet werden. Verwenden Sie bei den Nationen die Ländernamen als Objektbezeichner und geben Sie jeweils an, zu welcher Klasse die Objekte gehören.

20 b) In der Klasse SORTIERTELISTE sollen folgende Methoden implementiert werden:

- Eine Methode *anzahlAllerAthletenGeben()*, welche die Gesamtzahl der Athleten aller gespeicherten Nationen zurückgibt,
- eine Methode *ein fuegen(nationNeu)*, welche das übergebene Objekt *nationNeu* der Klasse NATION in die aktuelle Liste sortiert einfügt.

(Fortsetzung nächste Seite)

BE

Notieren Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung aller betroffenen Klassen der Listenstruktur. Wenden Sie soweit wie möglich das Prinzip der Rekursion an. Auf Attribute und Methoden, die zur Lösung nicht benötigt werden, kann verzichtet werden.

Zudem darf vorausgesetzt werden, dass in der Klasse NATION bereits eine Methode *istKleiner(nation)* existiert, die genau dann den Wahrheitswert *wahr* zurückgibt, wenn der eigene Landesbezeichner alphabetisch vor dem Landesbezeichner der übergebenen Nation steht.

- 6 c) Tatsächlich gibt es in der Reihenfolge der Nationen beim Einzug zwei Ausnahmen: Die Mannschaft Griechenlands marschiert zur Erinnerung an die antike Tradition stets als erste ins Stadion; die Mannschaft des Gastgeberlandes bildet immer den Schluss.
- Stellen Sie kurz eine geeignete Strategie dar, wie dieser Besonderheit Rechnung getragen werden kann. Nennen Sie die Konsequenzen Ihrer Strategie für die in Teilaufgabe 1b genannten Methoden.

- 6 2. Während des Wettkampfes ist eine weitere verkettete Liste aller beteiligten Nationen im Einsatz, welche es jederzeit ermöglicht, den aktuellen Medaillenspiegel auszugeben. Dabei werden alle Nationen nach ihrer Position sortiert: Die erste Position hat das Land mit der größten Anzahl an gewonnenen Goldmedaillen. Bei gleich vielen Goldmedaillen wird nach der Anzahl der Silbermedaillen, bei gleich vielen Gold- und Silbermedaillen nach der Anzahl der gewonnenen Bronzemedaillen sortiert. Bei gleichem Medaillenstand ist die Reihenfolge willkürlich.

Der aktuelle Medaillenstand einer Nation, also die jeweilige Anzahl der Gold-, Silber- und Bronzemedaillen, wird in einem Feld *medaillen* der Länge 3 mit ganzzahligen Elementen gespeichert.

Notieren Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache eine Vergleichsmethode *istBesser(nation)* der Klasse NATION, die genau dann den Wahrheitswert *wahr* zurückgibt, wenn die eigene Nation in ihrer Position vor der übergebenen Nation steht.

(Fortsetzung nächste Seite)

BE

3. Die Daten aller an der Olympiade beteiligten Athleten werden in einer eigenen Datenstruktur gespeichert, und zwar in einem geordneten Binärbaum. Sortiert wird nach einem Schlüssel, der eine Kombination aus Länderzugehörigkeit und fortlaufender Nummer der Form BB-zzz darstellt. Dabei bezeichnet B einen Großbuchstaben und z eine Ziffer (Beispiel: DE-322).

7 a) Ein geordneter Binärbaum enthält nur die Daten der in folgender Tabelle aufgeführten acht Athleten. Zeichnen Sie einen für die Suche möglichst effizient aufgebauten Binärbaum.

Geben Sie außerdem eine mögliche Einfügereihenfolge für den von Ihnen gezeichneten Binärbaum an.

Schlüssel	Name	Sportart	Nation
CH-003	Claudio Fürst	Schwimmen	Schweiz
CH-023	Karel Bürgi	Turnen	Schweiz
CH-554	Lena Leicht	Volleyball	Schweiz
DE-322	Franz Dehm	Volleyball	Deutschland
DE-400	Brigitte Stark	Schwimmen	Deutschland
DE-588	Nadja Kraus	Reiten	Deutschland
DE-700	Heike Bauer	Reiten	Deutschland
FR-002	Alain Dubois	Schwimmen	Frankreich

6 b) An der Olympiade 2012 in London nehmen etwa 12000 Athleten teil. Schätzen Sie rechnerisch ab, wie lange es maximal dauert, bis ein bestimmter Athlet gefunden wird, wenn eine wie in Teilaufgabe 3a verwendete optimale Baumstruktur gegeben ist.

Geben Sie an, wie lange dagegen ein Suchvorgang bei ungünstiger Baumstruktur maximal dauert. Sie dürfen hierbei annehmen, dass ein Vergleich beim Suchen 20 Nanosekunden ($= 0,2 \cdot 10^{-7}$ Sekunden) in Anspruch nimmt.

Vergleichen und bewerten Sie kurz die beiden Ergebnisse.

6 c) In der Klassenstruktur des geordneten Binärbaums sind die nachfolgend dargestellten Methoden *k1* und *k2* definiert. Beschreiben Sie allgemein, welche Ausgabe der Aufruf der Methode *k1* erzeugt.

(Fortsetzung nächste Seite)

BE

Klasse ATHLETENBAUM

BAUMELEMENT wurzel

...

Methode k1()

wurzel.k2()

endeMethode

endeKlasse

abstrakte Klasse BAUMELEMENT

...

abstrakte Methode k2()

endeKlasse

Klasse DATENKNOTEN erbt von BAUMELEMENT

BAUMELEMENT naechsterLinks

BAUMELEMENT naechsterRechts

ATHLET inhalt

Methode k2()

naechsterLinks.k2()

wenn (inhalt.nationGeben() ist gleich "Deutschland")

gib Text aus: inhalt.nameGeben() ": " inhalt.sportartGeben()

endeWenn

naechsterRechts.k2()

endeMethode

endeKlasse

Klasse ABSCHLUSS erbt von BAUMELEMENT

Methode k2()

endeMethode

endeKlasse

3

- d) Die unter Teilaufgabe 3a angegebenen Daten sind als Tabelle *athlet* mit den angegebenen Spaltenbezeichnern in einer relationalen Datenbank gespeichert. Geben Sie eine Datenbankabfrage (z. B. in SQL) an, mit der man eine Namensliste aller deutschen Athleten mit ihrer ausgeübten Sportart erhält.

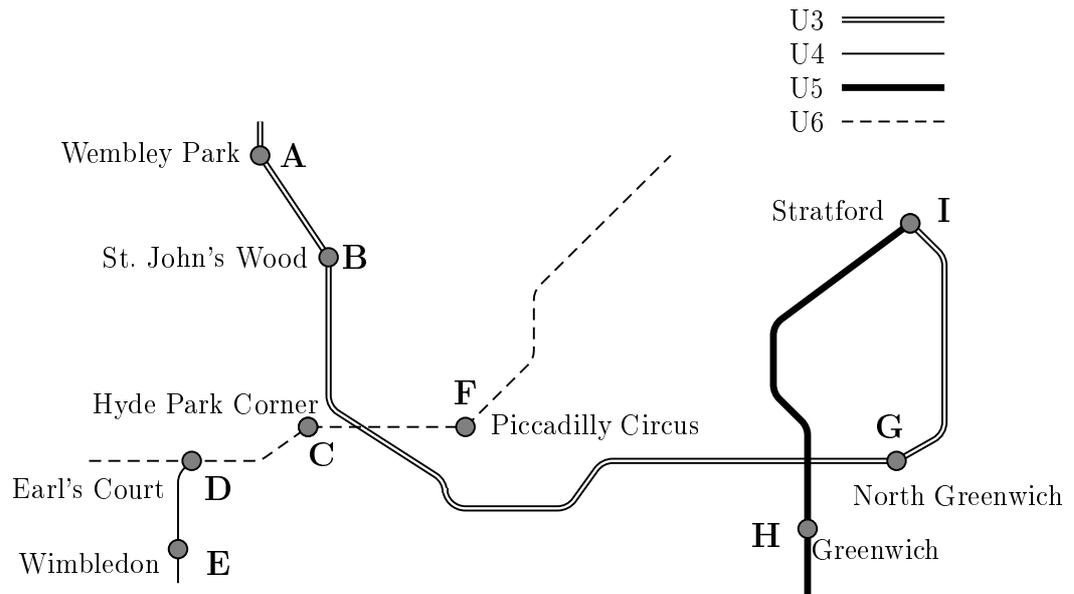
(Fortsetzung nächste Seite)

BE

4. Der unten abgebildete stark vereinfachte Ausschnitt aus dem U-Bahn-Netz Londons zeigt die mit den Bezeichnern U3 bis U6 versehenen Linien, die einige wichtige Spielstätten in London verbinden.

4

a) Repräsentieren Sie den gegebenen Ausschnitt des Linienplans als gewichteten Graph mit den angegebenen Buchstaben als Knotenmarkierungen und der jeweiligen U-Bahn-Nummer als Kantengewicht.



4

b) Erstellen Sie die zu Teilaufgabe 4a gehörige Adjazenzmatrix. Ordnen Sie die Bezeichner der Zeilen bzw. Spalten alphabetisch.

2

c) Entscheiden Sie, ob sich der gegebene Graph als Baum darstellen lässt. Begründen Sie Ihre Antwort kurz.

(Fortsetzung nächste Seite)

BE

10

5. Die Softwarefirma OnlineApps Inc. wird beauftragt, für den Kauf von Karten einzelner Veranstaltungen der Olympiade 2012 ein Online-Portal zu entwickeln, so dass über das Internet etwa 400000 Tickets verkauft werden können. Dabei soll eine limitierte Anzahl von Tickets für die Wettkämpfe von 17 verschiedenen Sportarten online vertrieben werden, z. B. 9500 Tickets für Schwimmen, 51300 für Leichtathletik usw.

Die Entwickler der Software müssen sich an folgende Vereinbarungen halten:

Wird das Online-Portal über einen Internet-Browser aufgerufen, so kann sich der Kunde zunächst entweder alle Veranstaltungen eines bestimmten Tages oder alle Veranstaltungen einer bestimmten Sportart anzeigen lassen. Hier kann der Kunde jeweils eine konkrete Veranstaltung auswählen, wodurch eine weitere Seite mit genaueren Angaben zum Wettkampfevent (Austragungsort, Termin und Anzahl noch verfügbarer Tickets) angezeigt wird. Falls nun die Anzahl der vom Kunden im nächsten Schritt eingegebenen Tickets noch vorrätig ist, wird eine weitere Seite zur Eingabe persönlicher Daten für die Kaufabwicklung geladen. Durch Bestätigung wird der Online-Kauf abgeschlossen und die Startseite des Online-Portals erneut geladen. Vor der endgültigen Bestätigung soll der Kunde die Kaufabwicklung jederzeit abbrechen können, was dazu führen soll, dass die Startseite des Online-Portals erneut in den Browser geladen wird.

Erstellen Sie ein Zustandsdiagramm für die Anwendersicht des Online-Portals, welches die beschriebenen Vorgaben berücksichtigt.

80

INF1. MODELLIERUNG UND PROGRAMMIERUNG

II.

BE

Der Betreiber der Mensa eines Gymnasiums stellt auf bargeldlose Bezahlung um. Zu diesem Zweck ersetzt er die bisherige Kasse durch ein Kassensystem mit integriertem Kartenleser.

Zur Teilnahme an diesem System werden von den Kunden Name, Vorname und Geburtsdatum erfragt und zusammen mit einer eindeutigen, automatisch generierten, ganzzahligen Kundennummer im Kassensystem gespeichert. Zusätzlich erhält der Kunde eine Chipkarte ausgehändigt, auf der ebenfalls die Kundennummer gespeichert ist. Um mit der Karte bezahlen zu können, muss ein genügend großes Guthaben vorhanden sein, welches zusammen mit den Kundendaten im Kassensystem gespeichert ist.

Bei der Bezahlung identifiziert sich der Kunde mit seiner Chipkarte am Kassensystem. Dann wird der Gesamtbetrag des Einkaufs ermittelt und zusammen mit der Einkaufszeit abgespeichert. Zusätzlich wird das gespeicherte Guthaben des Kunden um den Gesamtbetrag des Einkaufs reduziert. Jedem Kunden wird nur sein letzter Einkauf zugeordnet.

Um das Guthaben zu erhöhen, identifiziert sich der Kunde ebenfalls mit der Chipkarte am Kassensystem und zahlt dann den gewünschten Betrag ein, der zu seinem Guthaben addiert wird.

Damit jederzeit der aktuelle Tagesumsatz abgerufen werden kann, wird ein entsprechendes Attribut nach jedem Einkauf automatisch aktualisiert.

Hinweis: Für die oben genannte Aufzeichnung der Einkaufszeit wird ein Attribut *einkaufszeit* verwendet, in dem die Anzahl der seit dem 1.9.2011 00:00 Uhr vergangenen Minuten gespeichert wird.

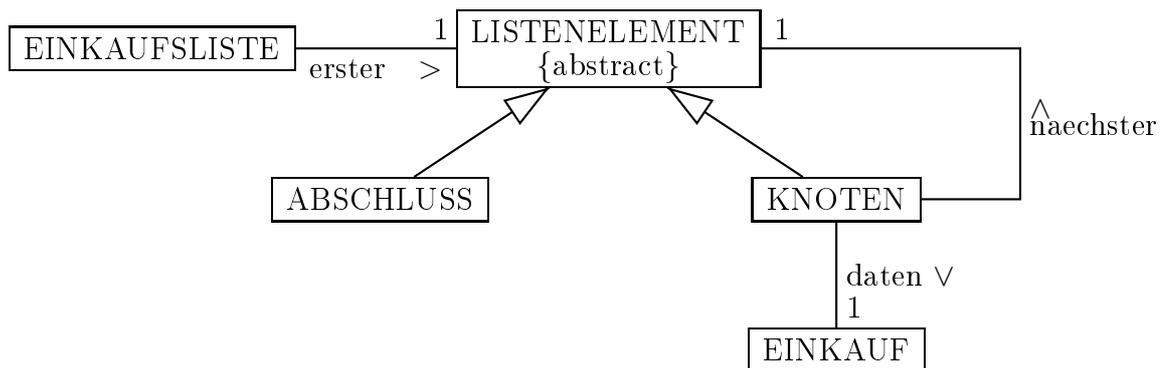
9

1. Erstellen Sie ein Klassendiagramm für das beschriebene Kassensystem unter Verwendung der Klassen KUNDE, KASSENSYSTEM und EINKAUF.

(Fortsetzung nächste Seite)

BE

2. Auf Wunsch des Schulforums soll ein Kunde nicht nur den letzten Einkauf abrufen, sondern am Monatsende alle seine Einkäufe und den Gesamtbetrag der Einkäufe einsehen können. Dazu werden für jeden Kunden die einzelnen Einkäufe in einer einfach verketteten Liste gespeichert. Ein Modell der einfach verketteten Liste ist im Wesentlichen aus folgendem Klassendiagramm ersichtlich:



- 6 a) Die Einkäufe könnte man auch mithilfe eines Feldes verwalten. Nennen Sie allgemein zwei wesentliche Vorteile der Verwendung einer einfach verketteten Liste gegenüber der Verwendung eines Feldes und bewerten Sie, inwiefern diese im gegebenen Anwendungsfall zum Tragen kommen.
- 11 b) Geben Sie in einer auf dem Deckblatt angegebenen Programmiersprache unter Verwendung des Prinzips der Rekursion eine mögliche Implementierung aller notwendigen Methoden an, mit denen ein Objekt der Klasse **EINKAUFSLISTE** die Summe der Beträge aller Einkäufe berechnen und zurückgeben kann. Hierbei können Sie auf die Standardmethode *gesamtbetragGeben()* der Klasse **EINKAUF** zurückgreifen.
- 3 c) Beschreiben Sie – beispielsweise mithilfe eines Klassendiagramms – eine Möglichkeit, Artikelbezeichner und Preis der Einzelposten, die zu einem Einkauf gehören, zu speichern.

(Fortsetzung nächste Seite)

BE

3. Ein Kunde soll vor Beginn des Einkaufs sein Guthaben prüfen können. Dazu werden Lesegeräte aufgestellt, die mit dem Kassensystem verbunden sind. Die Kundendaten werden im Kassensystem mithilfe eines geordneten Binärbaumes gespeichert. Als Schlüssel dient hierbei die Kundennummer. Bei Auflegen der Chipkarte wird anhand der darauf gespeicherten Kundennummer der zugehörige Datensatz im Baum gesucht und das Guthaben ausgegeben.
- 4 a) Nennen Sie den Vorteil eines solchen Baumes gegenüber einer sortierten Liste.
Erläutern Sie diesen Vorteil an einem Beispiel mit 1200 Kunden. Dabei dürfen Sie einen ausbalancierten Binärbaum annehmen, d. h. einen Binärbaum mit möglichst wenig Ebenen.
- 14 b) Stellen Sie die Baumstruktur in einem Klassendiagramm unter Verwendung des Softwaremusters Kompositum dar. Auf die Angabe von Attributen und Methoden kann dabei verzichtet werden. Geben Sie zusätzlich in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung einer Methode *kundeSuchen(kundennummer)* an, die zu einer übergebenen Kundennummer eine Referenz auf das zugehörige Objekt der Klasse KUNDE zurückgibt, sowie aller dazu nötigen Methoden in den jeweiligen Klassen. Hierbei können Sie auf die Standardmethode *kundennummerGeben()* der Klasse KUNDE zurückgreifen. Wenden Sie soweit wie möglich das Prinzip der Rekursion an.
- 3 c) Zu Beginn des neuen Schuljahres werden 190 neue Schüler erfasst. Sie erhalten fortlaufende, aufsteigende Kundennummern, die an die größte bisher vergebene Kundennummer anschließen. Die erfassten Daten werden in dieser Reihenfolge in den Binärbaum einsortiert. Beschreiben Sie die dabei auftretende Besonderheit für den Baum und die daraus resultierende Folge für den in Teilaufgabe 3a beschriebenen Vorteil.
- 6 d) Beschreiben Sie einen Algorithmus, der die Kunden nach Kundennummer geordnet anzeigt. Geben Sie ausgehend von der Ordnung des von Ihnen zugrunde gelegten Binärbaumes an, ob durch Ihren Algorithmus die Kundennummern aufsteigend oder absteigend sortiert werden.

(Fortsetzung nächste Seite)

BE

4. Der Mensabetreiber beliefert mittags weitere Einrichtungen. Die möglichen direkten Verbindungswege und deren Länge sind der folgenden Adjazenzmatrix zu entnehmen:

von \ nach	Gymnasium (G)	Realschule (R)	Mittelschule (M)	Seniorenstift (S)	Kinderhort (K)
G	0	1	3		
R	1	0		5	3
M		3	0	4	2
S		5	4	0	3
K		4	1	2	0

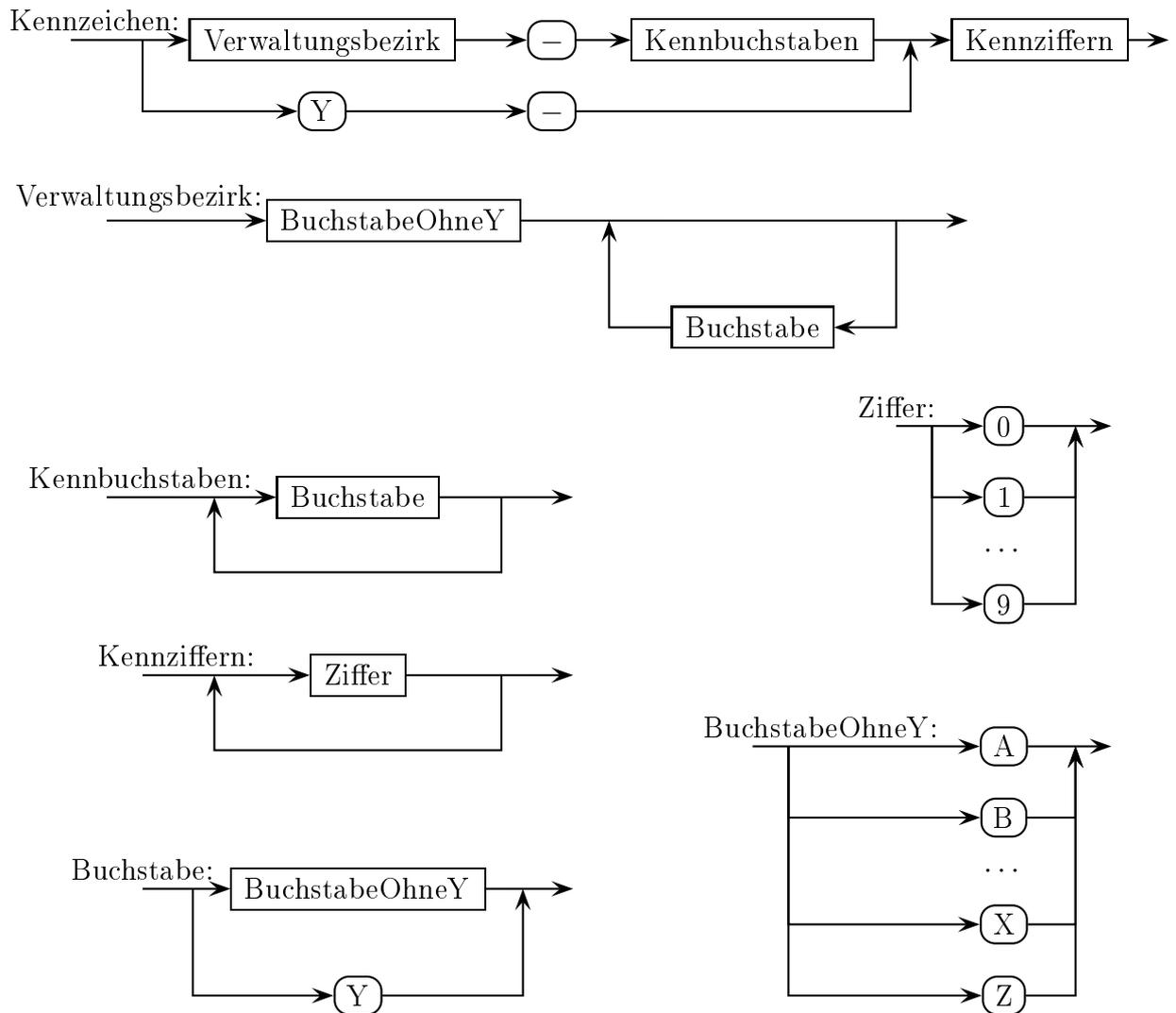
- 5 a) Stellen Sie die Situation als Graph dar und untersuchen Sie anhand der Adjazenzmatrix, ob der Graph ungerichtet ist.
- 3 b) Untersuchen Sie, ob der Graph im vorliegenden Fall als Baum dargestellt werden kann.
- 3 c) Geben Sie zwei verschiedene Pfade für den Weg vom Gymnasium zum Kinderhort an und bestimmen Sie die jeweilige Gesamtlänge des Weges.
- 4 d) Nennen Sie einen Algorithmus zum Graphendurchlauf und beurteilen Sie, ob dieser zur Bestimmung eines möglichst kurzen Weges geeignet ist.
5. Der Mensabetreiber möchte sein Kassensystem dahingehend erweitern, dass Kunden unter Verwendung der Chipkarte auf Kredit einkaufen können. Am Monatsende wird dann der geschuldete Betrag von einem vom Kunden angegebenen Konto eingezogen.
Eine Softwarefirma erhält den Auftrag zur Realisierung dieses Projektes.
- 6 a) Beschreiben Sie eine mögliche Strukturierung eines Projektablaufs in der Softwareentwicklung.
- 3 b) Erläutern Sie die Bedeutung von Meilensteinen für ein Softwareprojekt. Formulieren Sie einen möglichen Meilenstein für das beschriebene Projekt.

INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

III.

BE

1. In einer stark vereinfachten Form lässt sich die Syntax deutscher Autokennzeichen mit nachfolgendem Syntaxdiagramm beschreiben. An Sondernummern sind nur die mit "Y" beginnenden Nummern der Bundeswehrfahrzeuge vorgesehen.



4

- a) Geben Sie die Syntax der Sprache dieser Kennzeichen in formaler textueller Notation (z. B. erweiterter Backus-Naur-Form) an.

5

- b) Zeichnen Sie das Zustandsübergangsdiagramm eines erkennenden Automaten, der die Sprache der hier beschriebenen Kennzeichen akzeptiert.

(Fortsetzung nächste Seite)

BE

2. In einer Betriebskantine tragen vier Mitarbeiter die Tablettts der Mittagsgäste ab und stellen sie auf die Theke der Geschirrrückgabe. Ein weiterer Mitarbeiter räumt die Tablettts von der Theke auf das Förderband der Geschirrspülmaschine. Auf der Theke ist Platz für maximal 10 Tablettts.

Für die Simulation in einem Programm wird das Szenarium durch die Klassen TABLETT, ABTRAEGER, EINRAEUMER und THEKE modelliert.

Die Klasse THEKE hat als Attribute ein Feld mit Platz für 10 Objekte der Klasse TABLETT sowie einen Zähler für die Anzahl der Tablettts auf der Theke. Sie stellt nachstehende Methoden zur Verfügung:

- *ablegen(t)* legt das Tablett *t* wenn möglich auf der Theke ab; der Rückgabewert gibt an, ob das Tablett abgelegt werden konnte;
- *wegnehmen()* entfernt ein Tablett von der Theke und gibt seine Referenz zurück; falls kein Tablett vorhanden war, ist der Rückgabewert die leere Referenz.

Jeder der fünf simulierten Mitarbeiter agiert durch einen eigenen Thread (leichtgewichtiger Prozess) unabhängig von den anderen.

3 a) Geben Sie zwei typische Sondersituationen an, die sich für die Mitarbeiter ergeben können, und nennen Sie je eine mögliche Reaktion.

5 b) Zeigen Sie anhand eines geeigneten Sequenzdiagramms ein Problem auf, das bei einer (teilweise) parallelen Ausführung der Methode *ablegen* von zwei verschiedenen Threads aus auftreten könnte.

3 c) Erläutern Sie das Monitorkonzept und begründen Sie kurz, warum es zur Lösung des in Teilaufgabe 2b aufgezeigten Problems geeignet ist. Geben Sie zusätzlich auch an, wie sich Ihr Sequenzdiagramm durch den Einsatz eines Monitors verändert.

7 d) Notieren Sie eine Implementierung der Klasse THEKE in einer auf dem Deckblatt genannten Programmiersprache. Dabei sollen durch die Verwendung des Monitorkonzepts alle Synchronisationsprobleme verhindert werden.

(Fortsetzung nächste Seite)

BE

3. Gegeben ist eine Registermaschine mit folgendem Befehlssatz:

`load x` kopiert den Wert aus der Speicherzelle `x` in den Akkumulator
`loadi n` lädt den Wert `n` in den Akkumulator
`store x` kopiert den Wert des Akkumulators 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
`mul x` multipliziert den Wert aus der Speicherzelle `x` mit dem Wert im Akkumulator
`div x` dividiert den Wert im Akkumulator durch den Wert aus der Speicherzelle `x` (ganzzahlige Division)
`hold` beendet die Abarbeitung des Programms
`jmp x` springt zum Befehl in Speicherzelle `x`
`cmp x` vergleicht den Wert im Akkumulator mit dem Wert in Speicherzelle `x`. Das Ergebnis dieser Operation lautet +1 falls der Wert im Akkumulator größer ist als derjenige in Speicherzelle `x`, 0 bei Gleichheit und -1 sonst.
`jmpge x` springt zum Befehl in Speicherzelle `x`, wenn das Ergebnis der letzten Operation positiv oder 0 war
`jmpne x` springt zum Befehl in Speicherzelle `x`, wenn das Ergebnis der letzten Operation ungleich 0 war

Dabei bezeichnet `x` die Adresse einer Speicherzelle der Registermaschine; es gilt: $x \in \{0; 1; 2; \dots; 65535\}$.

2

a) In der Speicherzelle mit der Adresse 100 ist der Wert 13 gespeichert, in der Speicherzelle mit Adresse 101 der Wert 5. Geben Sie die Belegung der Speicherzellen 100, 101 und 102 nach Ausführung des folgenden, ab Adresse 0 gespeicherten Programms an.

```

load 100
div 101
mul 101
store 102
load 100
sub 102
store 102
hold
  
```

(Fortsetzung nächste Seite)

BE

2

b) Geben Sie an, welche Operation die in Teilaufgabe 3a beschriebene Befehlsfolge für beliebige Anfangsbelegungen der Zellen 100 und 101 ausführt.

Um zu testen, ob eine nicht durch 2 oder 3 teilbare natürliche Zahl eine Primzahl ist, kann man mit folgendem Algorithmus untersuchen, ob es andere Teiler gibt:

Zahl = zu untersuchender Wert

Divisor = 5

HatTeiler = 0

Erhoehung = 2

wiederhole solange Divisor < Zahl

wenn $(\text{Zahl} / \text{Divisor}) * \text{Divisor}$ ist gleich Zahl

dann

HatTeiler = 1

endewenn

Divisor = Divisor + Erhoehung

wenn Erhoehung ist gleich 2

dann

Erhoehung = 4

sonst

Erhoehung = 2

endewenn

endewiederhole

Hinweise: Die abwechselnde Erhöhung des Divisors um 2 oder 4 bewirkt, dass Vielfache von 2 und 3 nicht mehr getestet werden. „/“ bedeutet eine ganzzahlige Division.

1

c) Geben Sie die Bedeutung der Bedingung

$(\text{Zahl} / \text{Divisor}) * \text{Divisor}$ ist gleich Zahl

in der sechsten Zeile des Algorithmus an.

8

d) Schreiben Sie ein Programm für die gegebene Registermaschine, das den obigen Algorithmus umsetzt. Geben Sie dabei gegebenenfalls auch an, welche Speicherzellen zu Beginn der Programmausführung welchen Wert besitzen müssen.

40

INF2. THEORETISCHE UND TECHNISCHE INFORMATIK

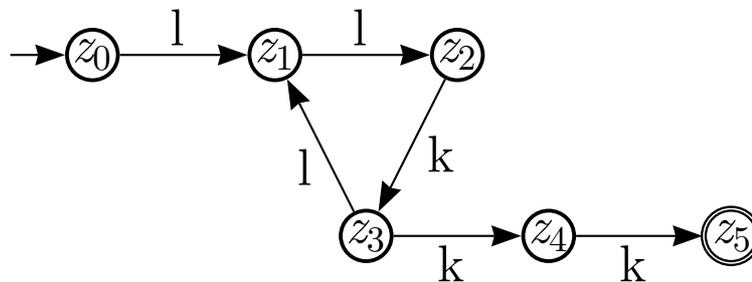
IV.

BE

1. Der Geheimdienst von Infotasia (GI) hat eine konspirative Wohnung angemietet. Geöffnet wird hier nur, wenn ein Besucher sich mit einem zulässigen Muster aus kurzen und langen Klingeltonen anmeldet. Alle zulässigen Muster sind durch den nachfolgend dargestellten, endlichen erkennenden Automaten gegeben.

k: kurzer Klingelton

l: langer Klingelton



- 3 a) Entscheiden Sie, ob die folgenden Muster zulässig sind. Begründen Sie Ihre Antwort kurz.

i) llkk

ii) llkllkkk

- 9 b) Notieren Sie eine Implementierung eines entsprechenden Automaten mithilfe einer auf dem Deckblatt angegebenen Programmiersprache. Dabei soll es u. a. eine Methode `codeTesten(wort)` geben, die überprüft, ob ein gegebenes Wort ein zulässiges Muster darstellt oder nicht, worauf sie einen entsprechenden Wahrheitswert zurückgibt. Bei den Zustandsübergängen können Sie sich exemplarisch auf die Übergänge aus z_3 und z_5 beschränken.

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.

(Fortsetzung nächste Seite)

BE

5

- c) Der Besuchercode wird geändert. Die künftig erlaubten Klingelmuster werden in einer Textnotation folgendermaßen vereinbart:

$$\text{klingelmuster} = \{kl\}l\{ll\}k ;$$

Dabei bedeutet $\{ \}$, dass der geklammerte Ausdruck beliebig oft – auch keinmal – wiederholt werden kann.

Entwerfen Sie ein Zustandsübergangsdiagramm eines endlichen Automaten über dem Alphabet $\{k, l\}$, der genau die erlaubten Klingelmuster erkennt.

4

2. Bei der Von-Neumann-Architektur wird der Speicher des Rechners gemeinsam von Programmen und Daten genutzt.

Beschreiben Sie in jeweils einem Satz einen Vor- und einen Nachteil dieses Konzepts.

3. Die deutsche Bundespräsidentin O. Inf führt ein Telefongespräch mit dem slowenischen Regierungschef R. Matik. Da keiner die Sprache des anderen versteht und auch keine Dolmetscher in der jeweils anderen Sprache verfügbar sind, wird vereinbart, dass die Gesprächspartner über je einen englischsprechenden Dolmetscher kommunizieren. Aus Geheimhaltungsgründen wird das Telefongespräch zusätzlich digitalisiert und verschlüsselt.

6

- a) Stellen Sie die Kommunikation zwischen den Politikern mit Hilfe eines Schichtenmodells dar. Als unterste Schicht darf die bestehende Datenleitung dienen. Für jede Schicht soll deutlich werden, welche Aufgabe sie hat.

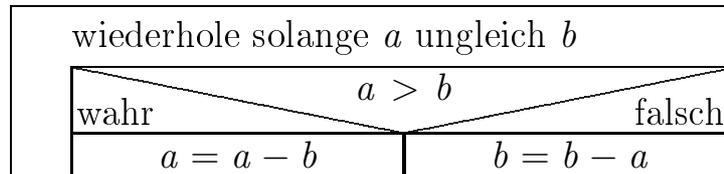
4

- b) Beschreiben Sie den zeitlichen Ablauf eines typischen Gesprächsausschnitts, bestehend aus einer Frage aus Deutschland und der Antwort aus Slowenien, durch ein Sequenzdiagramm, welches sich auf die beteiligten Personen als Objekte beschränkt.

(Fortsetzung nächste Seite)

BE

4. Das nachstehende Struktogramm beschreibt einen einfachen Algorithmus:



- 3 a) Machen Sie anhand einer Belegungstabelle deutlich, welche Werte die Variablen a und b nach Ablauf des Algorithmus haben, wenn sie zu Beginn die Werte $a = 14$ und $b = 8$ hatten.
- 6 b) Übertragen Sie den Algorithmus in ein Programm für eine Registermaschine mit nachfolgendem Befehlssatz. Machen Sie deutlich, in welchen Speicherzellen die Werte von a und b zu Beginn und am Ende des Programmablaufs stehen.

loadi n	lädt die ganze Zahl n in den Akkumulator
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
inc	vergrößert den Wert im Akkumulator um 1
dec	verringert den Wert im Akkumulator um 1
jmp x	springt zum Befehl in Speicherzelle x
jg x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator positiv ist
je x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
jne x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator ungleich 0 ist
end	beendet die Abarbeitung des Programms