

Abiturprüfung 2025

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:

--

Bei der Bearbeitung der Aufgaben dürfen als Hilfsmittel verwendet werden:

- die vom Staatsministerium genehmigte Merkhilfe für das Fach Mathematik,
- eine der vom Staatsministerium für Leistungserhebungen zugelassenen naturwissenschaftlichen Formelsammlungen,
- ein Taschenrechner, der den vom Staatsministerium getroffenen Regelungen entspricht,
- ein Rechtschreibwörterbuch Deutsch, das nach Erklärung des Verlags die aktuellen amtlichen Regeln vollständig umsetzt.

Die Angabe ist vom Prüfling mit dem Namen zu versehen und abzugeben.

Name des Prüflings

BE

Für ein Skigebiet soll zur Verwaltung von Pisten, Liften, Hütten und Skipässen eine Software entwickelt werden. Dabei soll eine zentrale Klasse SYSTEM zum Einsatz kommen.

1. Zu jeder Piste des Skigebiets werden eine eindeutige Nummer, der Schwierigkeitsgrad und die Länge gespeichert. Jeder Lift des Skigebiets hat einen eindeutigen Namen. Außerdem sind für jeden Lift die Länge und die Fahrt-dauer relevant. Man unterscheidet bei den Liften zwischen Schleppliften, Sesselliften und Gondelseilbahnen. Neben den oben genannten Daten wird für Schlepplifte gespeichert, ob es sich um einen Schlepplift für Kinder handelt, für Sessellifte der Typ (beispielsweise Zweisitzer oder Viersitzer) und für Gondelseilbahnen die zulässige Anzahl der Personen pro Gondel.

Eine Piste kann über mehrere Lifte erreicht werden. Ein Lift wiederum kann zu mehreren Pisten führen.

Im Skigebiet gibt es Hütten, von denen ein eindeutiger Name und die Höhe über dem Meeresspiegel gespeichert werden. Eine Hütte befindet sich stets an einer bestimmten Piste; an einer Piste können mehrere Hütten liegen.

Jeder Skipass hat eine eindeutige Nummer.

- 10
- a) Erstellen Sie anhand dieser Angaben ein Klassendiagramm, das u. a. die Klassen SYSTEM, PISTE, LIFT, HÜTTE und SKIPASS enthält. Wenden Sie an geeigneter Stelle das Konzept der Vererbung an. Auf die Angabe von Methoden kann verzichtet werden.

- 2
- b) Erklären Sie, warum es in diesem Szenario von Vorteil ist, das Konzept der Vererbung zu nutzen.

- 2
- c) Im Skigebiet werden Tages-, Wochen- und Saisonkarten sowie Punktekarten angeboten. Bei letzteren wird abhängig vom benutzten Lift eine bestimmte Anzahl an Punkten abgebucht.

Begründen Sie, dass im beschriebenen Szenario zwei Unterklassen der Klasse SKIPASS ausreichen, um die unterschiedlichen Arten von Skipässen sinnvoll zu modellieren.

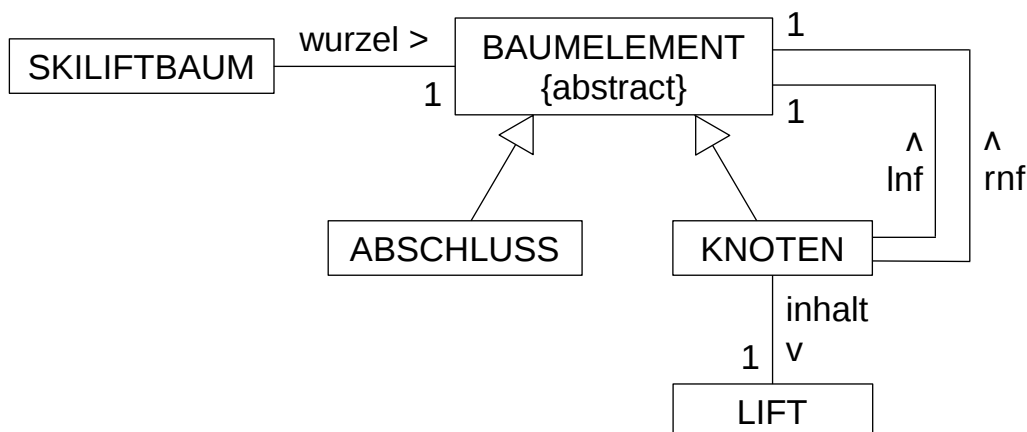
(Fortsetzung nächste Seite)

- d) Die Klasse SYSTEM verwaltet alle Piste-Objekte in einem Attribut *piste*, das als eindimensionales Feld implementiert wird. Gehen Sie davon aus, dass im Skigebiet mindestens eine Piste vorhanden ist, es aber auch leere Einträge im Feld geben kann.

Nun soll die durchschnittliche Länge aller Pisten des Skigebiets mithilfe einer Methode der Klasse SYSTEM ermittelt werden.

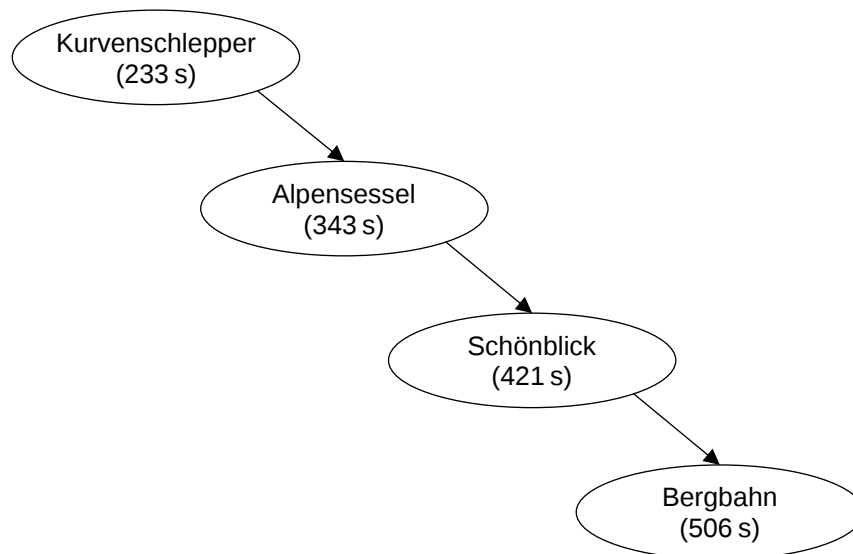
Notieren Sie dazu in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung der Methode *durchschnittslängeBestimmen*, die die durchschnittliche Länge aller Pisten zurückgibt. Die Klasse PISTE darf einschließlich der Methoden zum Geben der Attributwerte als vollständig implementiert vorausgesetzt werden.

2. Die Lift-Objekte werden vom System in einem geordneten Binärbaum gemäß dem folgenden Klassendiagramm verwaltet. Als Schlüssel dient die Fahrt-dauer, d. h., man geht davon aus, dass keine Fahrt-dauer mehrfach vorkommt. Dabei wird nicht zwischen den einzelnen Liftarten unterschieden.



(Fortsetzung nächste Seite)

- a) Im folgenden Baum sind die Knoten mit den Namen der Lifte und deren Fahrtauern beschriftet.



Beurteilen Sie, welche der drei bekannten Traversierungsarten für diesen speziellen Baum eine nach Fahrtdauer aufsteigend sortierte Ausgabe liefern.

Das Skigebiet verfügt über 63 Schlepplifte, 31 Sessellifte und 10 Gondelseilbahnen.

- 2 b) Ermitteln Sie, wie viele Ebenen ein Baum, in dem alle Lifte des Skigebiets verwaltet werden, mindestens haben muss.
- 3 c) In den zunächst leeren Binärbaum werden zuerst alle Schlepplifte, danach alle Sessellifte und zuletzt alle Gondelseilbahnen eingefügt. Dabei haben in diesem Skigebiet alle Sessellifte längere Fahrtauern als die Schlepplifte und kürzere als die Gondelseilbahnen.
- Beschreiben Sie den Aufbau des Baums, der dadurch entsteht.

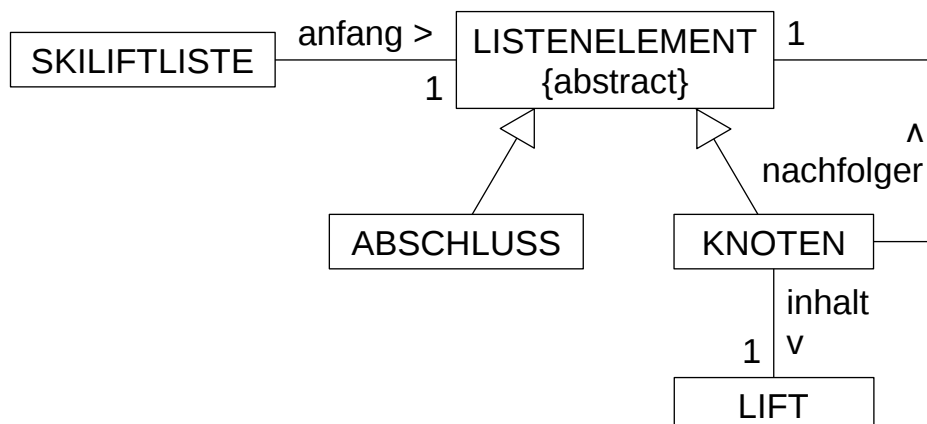
(Fortsetzung nächste Seite)

- d) Die Geschäftsführung hat den Wunsch geäußert, eine Übersicht über alle Lifte zu erhalten, die eine bestimmte Höchstfahrtdauer nicht überschreiten. Hierzu soll eine Methode *lifteAusgeben(höchstfahrtdauer)* zur Verfügung stehen, die alle Lifte aufsteigend nach Fahrtdauer sortiert ausgibt, deren Fahrtdauer höchstens dem Wert des Parameters *höchstfahrtdauer* entspricht.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *lifteAusgeben(höchstfahrtdauer)* in der Klasse SKILIFTBAUM und aller dazu nötigen Methoden in der Baumstruktur. Verwenden Sie so weit wie möglich das Prinzip der Rekursion und nutzen Sie die Sortierung des Baums. Sie können davon ausgehen, dass neben den Methoden zum Geben und Setzen von Attributwerten in der Klasse LIFT eine Methode *ausgeben()* existiert, die alle Daten des ausführenden Lift-Objekts auf dem Bildschirm ausgibt.

3. Jedes Mal, wenn ein Skipass an einem Lift gescannt wird, wird die Information, wann dieser Skipass an diesem Lift genutzt wurde, im System gespeichert. Im Zuge der Einführung der Software des Skigebiets wird auch eine Handy-App entwickelt, über die man u. a. nachvollziehen kann, welche Lifte man im Laufe eines Tages genutzt hat. Nach dem Erwerb eines Skipasses kann man dessen eindeutige Nummer in der App eingeben und so Zugriff auf die zum jeweiligen Skipass gehörenden Daten im System erhalten.

Die Handy-App verwendet zur Speicherung der genutzten Lifte eine einfach verkettete Liste gemäß dem folgenden Klassendiagramm. Wenn die Handy-App die Daten abrufen, werden die Lift-Objekte chronologisch eingefügt, wobei der zuletzt genutzte Lift am Anfang der Liste steht. Zu jeder Liftnutzung wird in der Liste das entsprechende Lift-Objekt referenziert.



(Fortsetzung nächste Seite)

4

- a) Der Nutzer eines Skipasses fährt nacheinander mit den Liften „Kurvenschlepper“, „Alpensessel“ und „Schönblick“. Stellen Sie die Liste der gefahrenen Lifte gemäß dem gegebenen Klassendiagramm in einem Objektdiagramm dar. Beschränken Sie sich bei den Attributen der Lift-Objekte auf den Namen des Lifts.

In der Testphase der App wurden einige Wünsche geäußert, die im Folgenden umgesetzt werden sollen.

6

- b) Da das Skigebiet 104 Lifte umfasst, verlieren manche Gäste den Überblick und wissen nicht mehr, an welchem Lift sie ihr Fahrzeug geparkt haben. Daher soll die Klasse SKILIFTLISTE eine rekursive Methode *startGeben* beinhalten, die eine Referenz auf das Lift-Objekt zurückgibt, das den Lift repräsentiert, der als erster benutzt wurde.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *startGeben* und aller dafür benötigten Methoden in den Klassen der Listenstruktur. Wurde noch kein Lift benutzt, soll eine leere Referenz zurückgegeben werden.

(Fortsetzung nächste Seite)

Es kommt immer wieder vor, dass Gäste persönliche Gegenstände (z. B. Autoschlüssel) im Skigebiet verlieren. Viele dieser Gegenstände werden von Findern bei den Liftstationen abgegeben.

Um den Betroffenen das Wiederfinden zu erleichtern, stellt die Klasse SKILIFTLISTE eine Methode *benutzteLifteAusgeben* zur Verfügung. Sie durchläuft alle gefahrenen Lifte und gibt deren Namen aus, beginnend bei dem zuletzt verwendeten Lift.

Wurde ein Lift mehrfach verwendet, soll dessen Name nur bei der letzten Verwendung des Lifts ausgegeben werden.

- 8 c) Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *benutzteLifteAusgeben* der Klasse SKILIFTLISTE und aller dazu nötigen Methoden in der Listenstruktur. Verwenden Sie so weit wie möglich das Prinzip der Rekursion.

Zur Implementierung der Methode steht Ihnen eine Klasse LIFTSCHLANGE mit den folgenden Methoden zur Verfügung:

- *einfügen(LIFT lift)* fügt das übergebene Lift-Objekt *lift* hinten ein.
- *enthält(ZEICHENKETTE name)* gibt genau dann *wahr* zurück, wenn ein Lift-Objekt in der Schlange enthalten ist, dessen Name mit dem übergebenen Namen *name* übereinstimmt.
- *ausgeben()* gibt die Namen der Lift-Objekte in der Schlange von vorne nach hinten auf dem Handydisplay aus.

Darüber hinaus existiert ein Konstruktor, der ein leeres Liftschlange-Objekt erzeugt.

Sie können ferner davon ausgehen, dass die Klasse LIFT einschließlich der Methoden zum Geben und Setzen der Attributwerte vollständig implementiert ist.

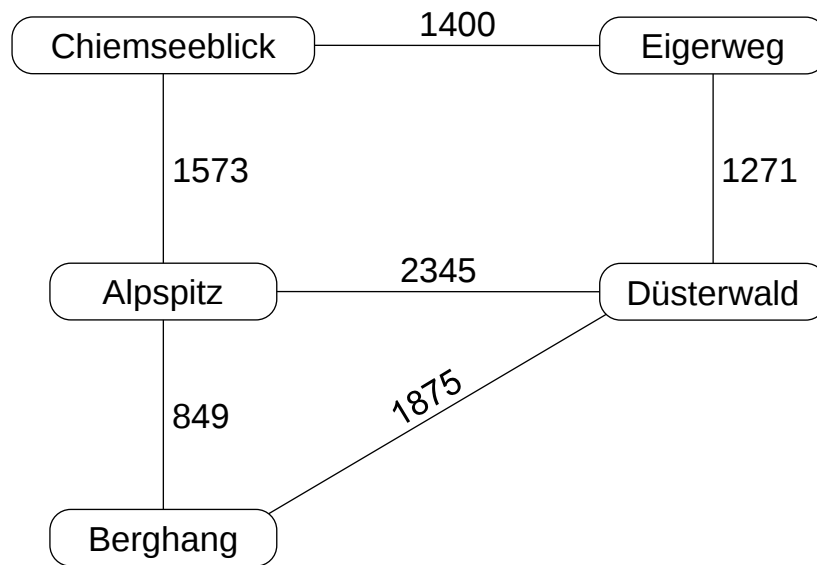
- 3 d) Erläutern Sie die Unterschiede zwischen den Datenstrukturen Stapel und Schlange.

Beurteilen Sie, ob man in Teilaufgabe 3c anstelle einer Schlange prinzipiell auch einen Stapel verwenden könnte.

(Fortsetzung nächste Seite)

4. Im System werden nicht nur die Hütten selbst repräsentiert, sondern auch, wie sie über Wirtschaftswege miteinander verbunden sind. Hierzu wird die Datenstruktur Graph verwendet. Dabei werden die Hütten als Knoten und die Wirtschaftswege als Kanten modelliert. Das Kantengewicht stellt die Entfernung zwischen den Hütten in Metern dar.

Folgender Graph zeigt eine Auswahl der Hütten des Skigebiets:



- 2 a) Erklären Sie, welche Bedeutung eine gerichtete Kante zwischen den Knoten „Berghang“ und „Düsterwald“ in obigem Graphen im Sachzusammenhang hätte. Beschreiben Sie eine mögliche Konsequenz, die sich daraus im Sachzusammenhang ergäbe.
- 4 b) Erstellen Sie die Adjazenzmatrix zu obigem Graphen.

Der Graph wird mithilfe einer Klasse GRAPH implementiert, in der die Knoten in einem Feld *knoten*, die Anzahl der Knoten im Attribut *anzahl* und die Adjazenzmatrix in einem zweidimensionalen Feld *matrix* gespeichert sind. In der Klasse KNOTEN existiert außerdem eine Methode *hüttennameAusgeben()*, die den Namen der zum Knoten gehörenden Hütte auf dem Bildschirm ausgibt.

(Fortsetzung nächste Seite)

- c) Die Bergwacht sucht einen neuen Standort an einer der Hütten des Skigebiets. Diese Hütte soll zentral liegen, damit man möglichst viele andere Hütten direkt erreichen kann.

Geben Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung einer Methode *bestenStandortAusgeben()* in der Klasse GRAPH an, die den Namen der Hütte mit den meisten direkten Nachbarn auf dem Bildschirm ausgibt. Falls es mehrere Hütten geben sollte, die sich als bester Standort eignen, soll nur eine davon ausgegeben werden.

Im Skigebiet gibt es zum einen Selbstversorgerhütten, zum anderen Hütten mit Gastronomie. Für die Lebensmittelversorgung der Hütten mit Gastronomie soll es die Möglichkeit geben, einen Graphen zu erzeugen, der nur diese Hütten als Knoten enthält.

Hierzu wird der gegebene Graph in einen neuen Graphen überführt, in dem keine Selbstversorgerhütten mehr enthalten sind. Kanten, die zu entsprechenden Hütten führen, müssen gegebenenfalls mit weiterführenden Kanten vereinigt werden, sofern nicht bereits kürzere Direktverbindungen existieren.

- d) Im oben gegebenen Graphen ist die Hütte „Alpspitz“ die einzige Selbstversorgerhütte. Zeichnen Sie den Graphen, der entsteht, wenn man den gegebenen Graphen in einen Graphen ohne diese Selbstversorgerhütte überführt.

- e) Formulieren Sie z. B. in Pseudocode einen Algorithmus für eine Methode *matrixAktualisieren*, der, wenn eine Selbstversorgerhütte außer Acht gelassen wird, die notwendigen Anpassungen an den Einträgen der Adjazenzmatrix vornimmt.

Gehen Sie dabei davon aus, dass der Graph weiterhin fünf Knoten enthält und der Index des Knotens zur einzigen Selbstversorgerhütte 0 ist. In der zugehörigen Adjazenzmatrix ist überall dort, wo keine Wege vorhanden sind, der Wert 0 eingetragen. Die Einträge in den Zeilen bzw. Spalten, die in der Adjazenzmatrix zur Selbstversorgerhütte gehören, sollen nach Ausführung der Methode den Wert 0 haben.

II.

BE

Ein Tanzstudio benötigt ein Verwaltungssystem und beauftragt ein Unternehmen, eine entsprechende Software zu entwickeln.

- 4 1. Nennen und beschreiben Sie kurz vier typische Phasen bei der Realisierung eines Softwareprojekts.

Die Software soll eine zentrale Klasse STUDIO enthalten, die Mitglieder, Kurse und Räume des Studios verwaltet.

Jedes Mitglied des Tanzstudios ist entweder Tanzschülerin bzw. Tanzschüler oder Tanzlehrkraft. Von jedem Mitglied sollen der Name, der Vorname und eine eindeutige Mitgliedsnummer gespeichert werden. Von Schülerinnen und Schülern wird zudem der Mitgliedsbeitrag erfasst, von Tanzlehrkräften dagegen ihr Grundgehalt.

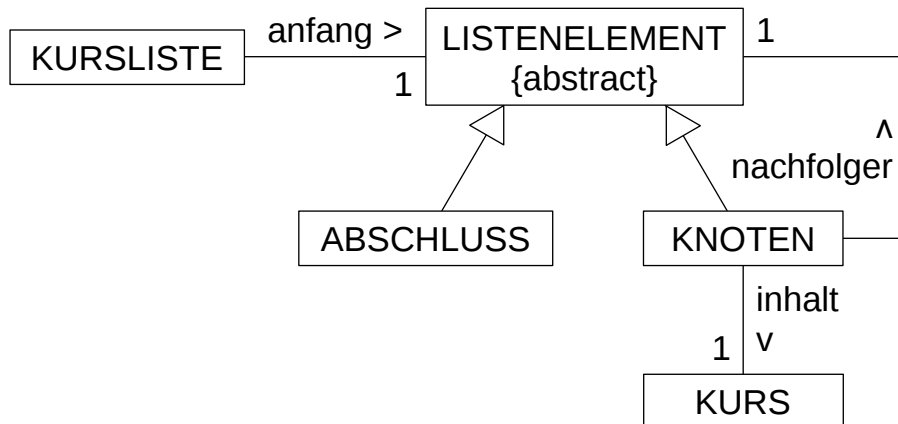
Jeder Tanzkurs hat einen eindeutigen Kursnamen und wird von genau einer Tanzlehrkraft geleitet. Schülerinnen und Schüler können sich für mehrere Kurse anmelden.

Die Räume des Tanzstudios haben eine eindeutige Raumnummer. Außerdem wird für jeden Raum die Kapazität gespeichert, die angibt, wie viele Schülerinnen und Schüler in diesem Raum höchstens unterrichtet werden können. Ein Tanzkurs findet immer im selben Raum statt.

- 9 2. Modellieren Sie das beschriebene Szenario in einem Klassendiagramm und setzen Sie dabei das Prinzip der Vererbung sinnvoll ein. Auf die Angabe von Methoden kann verzichtet werden.

(Fortsetzung nächste Seite)

3. Zur Verwaltung der angebotenen Tanzkurse wird eine einfach verkettete Liste gemäß dem folgenden Klassendiagramm verwendet.



Die Klasse **KURS** enthält neben dem Kursnamen ein ganzzahliges Attribut *niveau*, in dem das Schwierigkeitsniveau gespeichert wird. Außerdem gibt es die ganzzahligen Attribute *tag* und *startzeit*, die den Termin des Kurses angeben. Das Attribut *tag* nimmt dabei die Werte von 0 bis 6 für die Wochentage von Montag bis Sonntag an. Alle Tanzkurse finden einmal wöchentlich statt und dauern eine Stunde; sie beginnen immer zur vollen Stunde von 10:00 Uhr bis 21:00 Uhr. Das Attribut *startzeit* nimmt entsprechend Werte von 0 bis 11 an. Ein Kurs-Objekt hat darüber hinaus eine Referenz *raum* auf das Raum-Objekt, das den Raum repräsentiert, in dem der Kurs stattfindet.

In die zunächst leere Kursliste *kursliste* wurden die beiden Tanzkurse „Samba II“ (Termin: Mo, 17:00 Uhr; Niveau: 2) und „Charleston I“ (Termin: Do, 18:00 Uhr; Niveau: 1) nacheinander vorne eingefügt. Beide Kurse finden im Raum 17 statt, der Platz für 20 Schülerinnen und Schüler bietet.

- 5 a) Erstellen Sie ein Objektdiagramm, das den geschilderten Sachverhalt veranschaulicht.
- 8 b) Ein Schüler hat vergessen, in welchem Raum sein Tanzkurs „Samba II“ stattfindet. Für diesen Fall kann die Methode *raumnrGeben(kursname)* der Klasse **KURSLISTE** genutzt werden; sie gibt die Raumnummer des Raums zurück, in dem der Tanzkurs mit dem übergebenen Namen stattfindet. Stellen Sie grafisch z. B. mit einem Sequenzdiagramm einen passenden Ablauf nach Aufruf von *raumnrGeben("Samba II")* für die gegebene Kursliste *kursliste* dar. Berücksichtigen Sie alle beteiligten Objekte.

(Fortsetzung nächste Seite)

Die Entwickler der Software haben sich dazu entschieden, zur Verwaltung der Belegungspläne der Räume in der Klasse RAUM ein zweidimensionales Feld *belegung* zu verwenden. Dabei repräsentiert der erste Index mit Werten von 0 bis 6 den Wochentag und der zweite Index mit Werten von 0 bis 11 den Startzeitpunkt eines Tanzkurses.

In einem Feldelement kennzeichnet die Zeichenkette „frei“ ein Zeitfenster, in dem der zugehörige Raum noch nicht belegt ist, ansonsten ist dort der Name des Tanzkurses abgelegt, der während dieses Zeitfensters in diesem Raum stattfindet.

- 5 c) Häufig bieten Tanzlehrkräfte zwei Kurse direkt hintereinander an und möchten dabei im selben Raum bleiben. Dazu soll es eine Methode *freieDoppelstundenAusgeben* in der Klasse RAUM geben, die für ein Raum-Objekt alle freien Doppelstunden ausgibt. Implementieren Sie diese Methode in einer auf dem Deckblatt angegebenen Programmiersprache.

Sie können davon ausgehen, dass eine Methode *dsAusgeben(tag, startzeit)* der Klasse RAUM zur Verfügung steht, die den Termin derjenigen Doppelstunde auf dem Bildschirm ausgibt, die zu dem durch die übergebenen Werte von *tag* und *startzeit* festgelegten Zeitpunkt beginnt.

- 2 d) Der Termin eines Kurses wird einerseits beim dazugehörigen Kurs-Objekt und andererseits durch die Position in der Belegungstabelle des zugehörigen Raum-Objekts abgespeichert. Beschreiben Sie einen Vor- und einen Nachteil, der mit dieser Umsetzung verbunden ist.

- 9 e) In der Klasse KURSLISTE soll eine Methode *raumEintragen(kursname, raum)* zur Verfügung stehen, die in dem passenden Kurs-Objekt die übergebene Referenz *raum* im entsprechenden Attribut speichert. Beim Ablauf der Methode soll auch der entsprechende Wert im Attribut *belegung* des Objekts *raum* gesetzt werden. Falls in der Kursliste kein entsprechender Kurs existiert, soll *falsch* zurückgegeben werden, ansonsten *wahr*.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung der Methode *raumEintragen* in der Klasse KURSLISTE und aller dafür benötigten Methoden in den weiteren Klassen. Wenden Sie so weit wie möglich das Prinzip der Rekursion an. Sie können davon ausgehen, dass in der Klasse KURS die Methoden zum Geben und Setzen der Attributwerte bereits implementiert sind.

(Fortsetzung nächste Seite)

4. Zur Verwaltung der Mitglieder des Tanzstudios wird ein geordneter Binärbaum verwendet. Als Schlüssel dient dabei die Mitgliedsnummer.

Vereinfachend wird im Folgenden nicht zwischen Lehrkräften und Schülerinnen bzw. Schülern unterschieden. Sämtliche Mitglieder werden in einer Klasse MITGLIED verwaltet:

MITGLIED
GANZZAHL mitgliedsnummer
ZEICHENKETTE name
ZEICHENKETTE vorname
WAHRHEITSWERT istProfi
ZEICHEN geschlecht
datenAusgeben()

- 4 a) Zeichnen Sie einen für die Suche optimal aufgebauten Binärbaum, der die folgenden Mitglieder mit den in Klammern angegebenen Mitgliedsnummern enthält. Zur Vereinfachung genügt es, jeden Knoten durch die zugehörige Mitgliedsnummer zu bezeichnen.

Chris (111)

Jo (127)

Alex (311)

Dani (555)

Toni (618)

Kim (679)

Geben Sie zudem an, wie viele unterschiedliche Bäume für die gegebenen Mitglieder möglich sind, die für die Suche optimal aufgebaut sind. Begründen Sie Ihre Angabe.

- 2 b) Bei der Anmeldung im Tanzstudio werden neue Mitglieder unmittelbar nach Vergabe einer Mitgliedsnummer in den Binärbaum eingefügt. Wird hierbei eine ungünstige Vergabestrategie gewählt, entartet der Binärbaum zu einer sortierten Liste.

Geben Sie eine solche Vergabestrategie an. Beschreiben Sie eine alternative Vergabestrategie, die dieses Problem vermeidet.

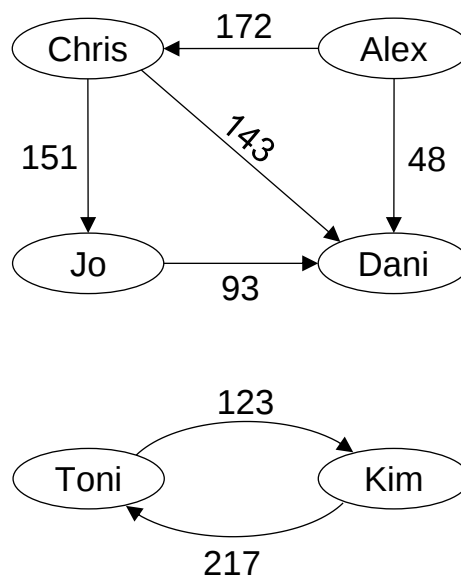
(Fortsetzung nächste Seite)

BE	
2	c) Berechnen Sie, wie viele Ebenen ein Baum mindestens haben muss, wenn die Mitgliederanzahl 300 beträgt. Bestimmen Sie für diesen Fall außerdem die Anzahl der freien Plätze.
6	d) Bei der Implementierung des geordneten Binärbaums zur Verwaltung der Mitglieder in der Klasse MITGLIEDERBAUM wurde das Entwurfsmuster Kompositum verwendet und die Trennung von Struktur und Daten berücksichtigt. Für eine Tanzaufführung des Tanzstudios werden Mitglieder gesucht, die als „Profi“ eingestuft sind. Es soll zudem möglich sein, das Geschlecht der gesuchten Mitglieder vorzugeben. Dazu soll eine Methode <i>profisAusgeben(geschlecht)</i> in der Klasse MITGLIEDERBAUM implementiert werden, welche die Daten aller gesuchten Mitglieder der Stufe „Profi“ mit dem angegebenen Geschlecht aufsteigend sortiert nach der Mitgliedsnummer am Bildschirm ausgibt. Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung dieser Methode für die Klasse MITGLIEDERBAUM und der dafür nötigen Methoden aller weiteren Klassen der Baumstruktur. Die Klasse MITGLIED kann einschließlich der Methoden zum Geben und Setzen der Attributwerte als vollständig implementiert vorausgesetzt werden. Dabei gibt die angegebene Methode <i>datenAusgeben</i> die Daten des Mitglieds formatiert aus.

(Fortsetzung nächste Seite)

5. Mitglieder des Tanzstudios nehmen regelmäßig an Turnieren im Paartanz teil. Dazu werden die gebildeten Tanzpaare in einem Graphen dargestellt. Bei den Paartänzen gibt es üblicherweise zwei verschiedene Rollen, einen „Leader“ sowie einen „Follower“. Im Graphen werden diese beiden Rollen durch Pfeile gekennzeichnet. Um zu bewerten, wie erfolgreich ein Tanzpaar ist, wird die Punktzahl, die das Tanzpaar in der entsprechenden Konstellation insgesamt bei Turnieren bereits erreicht hat, als Kantengewicht notiert.

Ein Ausschnitt eines Graphen für das Tanzstudio kann z. B. folgendermaßen aussehen:



Der Pfeil von Alex zu Chris stellt hier beispielsweise dar, dass Alex bei Turnieren die Rolle des Leaders und Chris die Rolle des Followers übernommen hat. In dieser Konstellation haben sie insgesamt 172 Punkte erreicht.

- a) Geben Sie eine Adjazenzmatrix an, die den gegebenen Graphen repräsentiert.
- b) In einem weiteren Tanzturnier erreichen die Paare (Jo, Dani) 65 Punkte und (Dani, Kim) 46 Punkte. Dabei übernehmen jeweils die Erstgenannten die Rolle des Leaders. Tragen Sie die dafür notwendigen Änderungen direkt in den oben abgebildeten Graphen ein.

(Fortsetzung nächste Seite)

In der Klasse GRAPH gibt es ein Attribut *anzahlKnoten*, in dem die Anzahl der Knoten des Graphen gespeichert wird. Außerdem werden die Knoten des Graphen im Feld *knoten* sowie die Adjazenzmatrix des Graphen im zweidimensionalen Feld *matrix* abgelegt. In der Klasse KNOTEN gibt es ein Attribut *inhalt*, welches das zugehörige Mitglied-Objekt referenziert.

Die Klasse MITGLIED kann so, wie sie in Aufgabe 4 gegeben ist, einschließlich der Methoden zum Geben und Setzen der Attributwerte als vollständig implementiert vorausgesetzt werden, ebenso die Klasse KNOTEN.

- 6 c) Zur Ermittlung des erfolgreichsten Mitglieds des Tanzstudios soll in der Klasse GRAPH eine Methode *erfolgreichstesMitgliedAusgeben* zur Verfügung stehen. Diese Methode soll den Namen desjenigen Mitglieds auf dem Bildschirm ausgeben, das in der Summe als Leader und Follower die meisten Punkte erreicht hat.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung dieser Methode. Im Falle eines Punktegleichstands soll der Name des zuerst gefundenen Mitglieds ausgegeben werden.

- 8 d) Es soll möglich sein, dass für ein bestimmtes Mitglied in seiner Rolle als Leader mögliche Follower vorgeschlagen werden. Dabei kommen alle Personen infrage, die bereits Follower dieses Mitglieds waren oder Follower von Followern des Mitglieds waren etc., wobei diese Ketten jeweils nur solange fortgesetzt werden, bis eine vorgegebene Mindestpunktzahl in der jeweiligen Konstellation unterschritten wird. Das heißt, es kommen alle Personen infrage, die im Graphen vom betrachteten Mitglied aus entlang von Kanten, die alle ein bestimmtes Mindestgewicht haben, erreichbar sind.

Formulieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine Methode *möglichePersonenAusgeben(start, mindestpunkte)* der Klasse GRAPH, die ausgehend vom Knotenindex *start* die Namen aller infrage kommenden Personen ausgibt. Sie können davon ausgehen, dass es in der Klasse KNOTEN ein boolesches Attribut *besucht* gibt.

(Fortsetzung nächste Seite)

- e) Beschreiben Sie die Vorgehensweise der folgenden Methode der Klasse GRAPH und interpretieren Sie ihren Rückgabewert beim Aufruf für einen Index i eines beliebigen Knotens im angegebenen Kontext.

Geben Sie außerdem den Rückgabewert beim Aufruf der Methode für obigen Graphen mit dem Index des Knotens „Chris“ an.

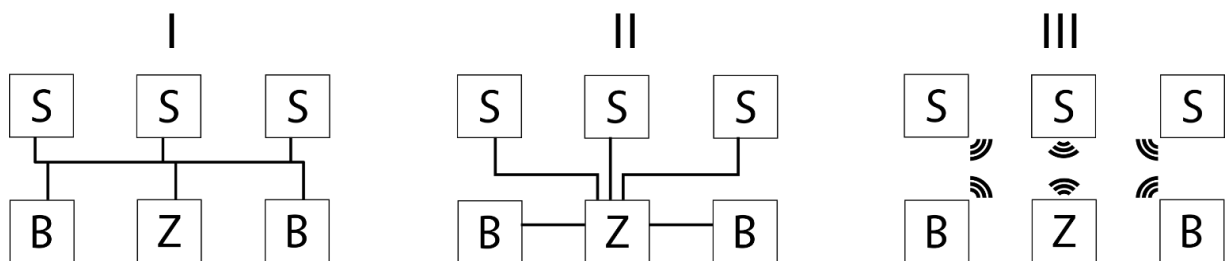
```
Methode methode(i)
    zahl = 0
    k = 0
    zähle j von 0 bis anzahlKnoten - 1
        wenn matrix[i][j] > zahl dann
            zahl = matrix[i][j]
            k = j
        endeWenn
    endeZähle
    gib knoten[k].name zurück
endeMethode
```

III.

BE

Das Alan-Turing-Gymnasium soll mit einer Alarmanlage ausgestattet werden. Diese besteht aus verschiedenen Sensoren (Bewegungsmeldern, Türkontakten etc.), einer Alarmzentrale mit Sirene sowie je einer Bedieneinheit außen am Haupteingang und in der Hausmeisterzentrale.

1. Damit die Alarmanlage funktioniert, müssen alle Sensoren (S) und Bedieneinheiten (B) mit der Alarmzentrale (Z) kommunizieren. Der Hersteller bietet drei Varianten des Alarmsystems an, die sich lediglich in der Art der Gerätevernetzung unterscheiden. Bei Variante I und II erfolgt die Datenübertragung über Kabel, bei Variante III per Funk.



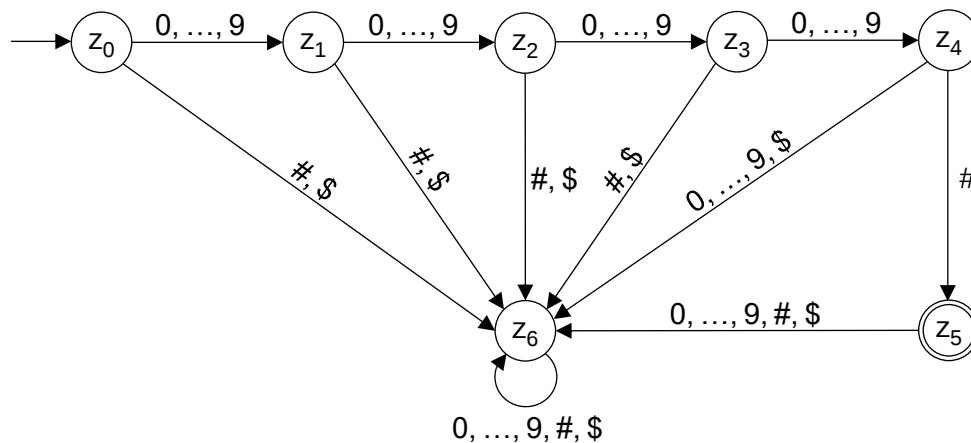
- 4 a) Geben Sie zu den in I und II gezeigten Topologien jeweils die Fachbegriffe an und beschreiben Sie einen Vor- und einen Nachteil der Variante II gegenüber Variante I.
- 2 b) Bei Variante III kommunizieren alle Geräte über einen gemeinsamen Funkkanal. Beurteilen Sie, welcher Topologie diese Variante am ehesten entspricht.
- 4 c) Beschreiben Sie anhand von zwei Gesichtspunkten, weshalb es für den Entwickler des Alarmsystems von Vorteil ist, wenn er sich bei der Realisierung der Kommunikation zwischen den Geräten an einem Schichtenmodell orientiert.

(Fortsetzung nächste Seite)

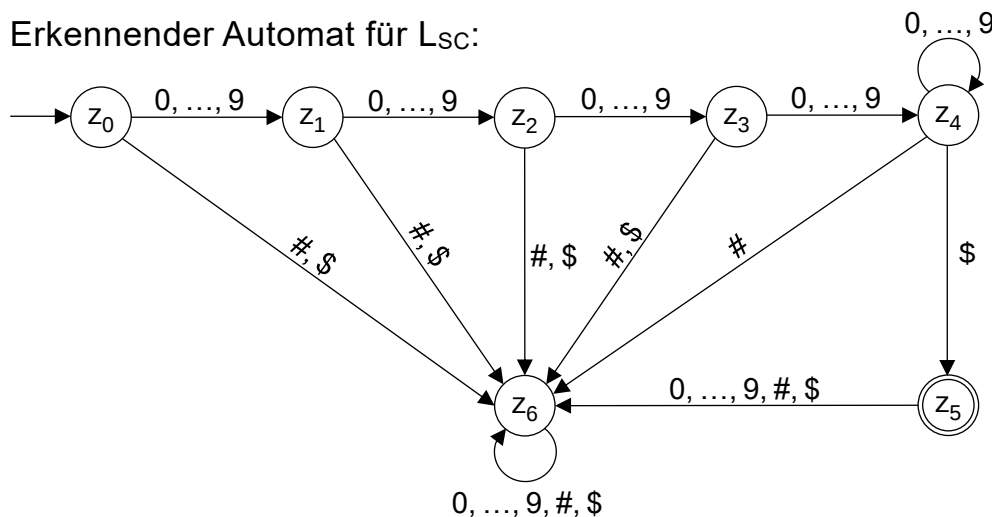
2. Die Steuerung der Alarmanlage erfolgt über spezielle Codes, die über die abgebildete Tastatur an den Bedieneinheiten eingegeben werden können. Dabei wird zwischen normalen Nutzercodes zum Aktivieren und Deaktivieren der Alarmanlage und speziellen Steuer-codes für die Konfiguration der Anlage unterschieden. Die Sprache L_{NC} der möglichen Nutzercodes wird vom oberen, die Sprache L_{SC} der möglichen Steuer-codes vom unteren der beiden abgebildeten endlichen Automaten erkannt.

1	2	3
4	5	6
7	8	9
#	0	\$

Erkennender Automat für L_{NC} :



Erkennender Automat für L_{SC} :



- a) Formulieren Sie jeweils in eigenen Worten, wie gültige Nutzer- bzw. Steuer-codes aufgebaut sind.

(Fortsetzung nächste Seite)

7

- b) Implementieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache den oberen der beiden abgebildeten Automaten, der die Sprache L_{NC} erkennt. Beschränken Sie sich bei der Implementierung der Zustandsübergänge exemplarisch auf die von den Zuständen z_4 und z_5 ausgehenden.

Sie dürfen dabei folgende Methoden der Klasse ZEICHENKETTE verwenden:

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

2

- c) Diskutieren Sie, ob es im Sachzusammenhang syntaktisch korrekte Steuer codes ohne Semantik geben kann.

4

- d) Geben Sie eine Grammatik für die Sprache L_C an, die sowohl alle gültigen Nutzer codes als auch alle gültigen Steuer codes umfasst. Stellen Sie dabei die Produktionsregeln in formaler Textnotation dar (z. B. in EBNF).

5

- e) Die Alarmanlage wird deaktiviert, wenn ein gültiger Nutzer code eingetippt oder eine kontaktlos auslesbare Chipkarte vor die Bedieneinheit gehalten wird. Die Chipkarte sendet dabei einen 64 Bit langen Schlüssel an die Bedieneinheit, der als 64-stellige Folge von Nullen und Einsen aufgefasst werden kann.

Eine Einbrecherin möchte die Alarmanlage unerlaubt deaktivieren. Sie kennt den Aufbau der Nutzer codes bzw. Chipkartenschlüssel und weiß, dass in der Alarmanlage 100 gültige Nutzer codes und zwei gültige Chipkartenschlüssel angelegt sind. Für das Ausprobieren eines Nutzer codes benötigt sie zwei Sekunden; die Chipkartenschlüssel kann sie dagegen mit ihrem Smartphone automatisiert durchprobieren, sodass hier nur 0,1 Sekunden für das Ausprobieren eines Schlüssels benötigt werden.

Berechnen Sie für beide Fälle, wie lange ein Brute-Force-Angriff auf die Bedieneinheit der Alarmanlage im schlechtesten Fall jeweils dauert. Bewerten Sie anhand Ihrer Ergebnisse die Sicherheit der Alarmanlage.

(Fortsetzung nächste Seite)

3. Ein Bewegungsmelder der Alarmanlage verfügt intern über zwei Detektormodule, ein Auswertungs- und ein Kommunikationsmodul, die auf einen gemeinsamen Speicher zugreifen.

- Immer dann, wenn Detektormodul 1 bzw. 2 eine Bewegung im Raum erkennt, setzt es den Wert der Speicherzelle 101 bzw. 102 auf den Wert 255, anderenfalls setzt es den Wert der jeweiligen Speicherzelle auf 0.
- Das Auswertungsmodul prüft kontinuierlich in einer Endlosschleife den Inhalt der Speicherzellen 101 und 102 und schreibt nur dann den Wert 255 in die Speicherzelle 103, falls in beiden Speicherzellen ein Wert ungleich 0 gespeichert ist. Dies soll Fehlalarme verhindern.
- Das Kommunikationsmodul prüft regelmäßig, ob in der Speicherzelle 103 ein Wert ungleich 0 hinterlegt ist; falls ja, meldet es den Alarm an die Alarmzentrale weiter und setzt den Inhalt dieser Speicherzelle zurück auf 0.

4

a) Schreiben Sie ein Programm für das Auswertungsmodul, das die oben beschriebene Funktionalität umsetzt. Gehen Sie davon aus, dass zum Programmstart alle Speicherzellen mit 0 vorbelegt sind und das Auswertungsmodul nach dem Prinzip einer Registermaschine arbeitet, die ausschließlich die folgenden Befehle unterstützt:

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
jmp x	springt zum Befehl in Speicherzelle x
jeq x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
jge x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator positiv oder gleich 0 ist
jle x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator negativ oder gleich 0 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
hold	beendet die Abarbeitung des Programms

(Fortsetzung nächste Seite)

Alle Module des Bewegungsmelders arbeiten zeitlich parallel. Der Zugriff auf einzelne Speicherzellen ist jedoch pro Speicherzelle stets nur exklusiv für ein Modul möglich. Falls mehrere Module gleichzeitig auf eine Speicherzelle zugreifen möchten, wird einem zufällig ausgewählten Modul der Zugriff gewährt. Die übrigen Zugriffsoperationen werden automatisch so lange verzögert, bis der Zugriff wieder möglich ist.

2

b) Beurteilen Sie, ob es durch die Zugriffe auf die Speicherzellen 101, 102 und 103 zu einer Verklemmung kommen kann.

4

c) Vereinzelt kann es vorkommen, dass der Bewegungsmelder keinen Alarm an die Zentrale meldet, obwohl für einen gewissen Zeitraum beide Detektormodule eine Bewegung erkannt haben und die Speicherzellen 101 und 102 jeweils mit dem Wert 255 belegt wurden.

Geben Sie einen Ablauf an, bei dem es zu einem solchen fehlerhaften Verhalten des Bewegungsmelders kommt. Übernehmen Sie hierzu die folgende Tabelle auf Ihren Antwortbogen und vervollständigen Sie diese.

Modul	Operation	Speicherzelle
Detektor 1	schreibt Wert 255	101
...	...	...

IV.

BE

Ein kleines Online-Versandhaus für Fahrradzubehör hält seine Artikel in einem Lager vorrätig. Ein bestellter Artikel wird aus dem Lager geholt und in die Versandabteilung gebracht. Dort wird er verpackt, adressiert und versandt.

1. Jeder Artikel erhält ein eindeutiges Kennzeichen für seine Position im Lager (Lagerkennzeichen). Es setzt sich aus der Regalreihe, der Regalnummer und dem Regalfach zusammen.

Es gibt acht Regalreihen, die mit den Buchstaben A bis H beschriftet sind. Die Regalreihen A bis D bestehen aus jeweils zehn Regalen (01 bis 10), die restlichen Regalreihen aus jeweils zwölf (01 bis 12). Jedes Regal hat sechs Fächer (Regalfach 1 bis 6).

Zum Beispiel ist B073 ein korrektes Lagerkennzeichen.

Die Menge aller derartigen Lagerkennzeichen bildet die formale Sprache L über dem Alphabet {A, B, ..., H, 0, 1, ..., 9}.

- 5 a) Zeichnen Sie das Zustandsübergangsdiagramm eines erkennenden endlichen Automaten, der genau die Wörter der Sprache L akzeptiert.

- 7 b) Implementieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache Ihren Automaten aus Teilaufgabe 1a. Dabei soll es u. a. eine Methode *prüfen(lagerkennzeichen)* geben, die überprüft, ob die übergebene Zeichenkette *lagerkennzeichen* den Vorgaben für die Lagerkennzeichen entspricht, und den entsprechenden Wahrheitswert zurückgibt.

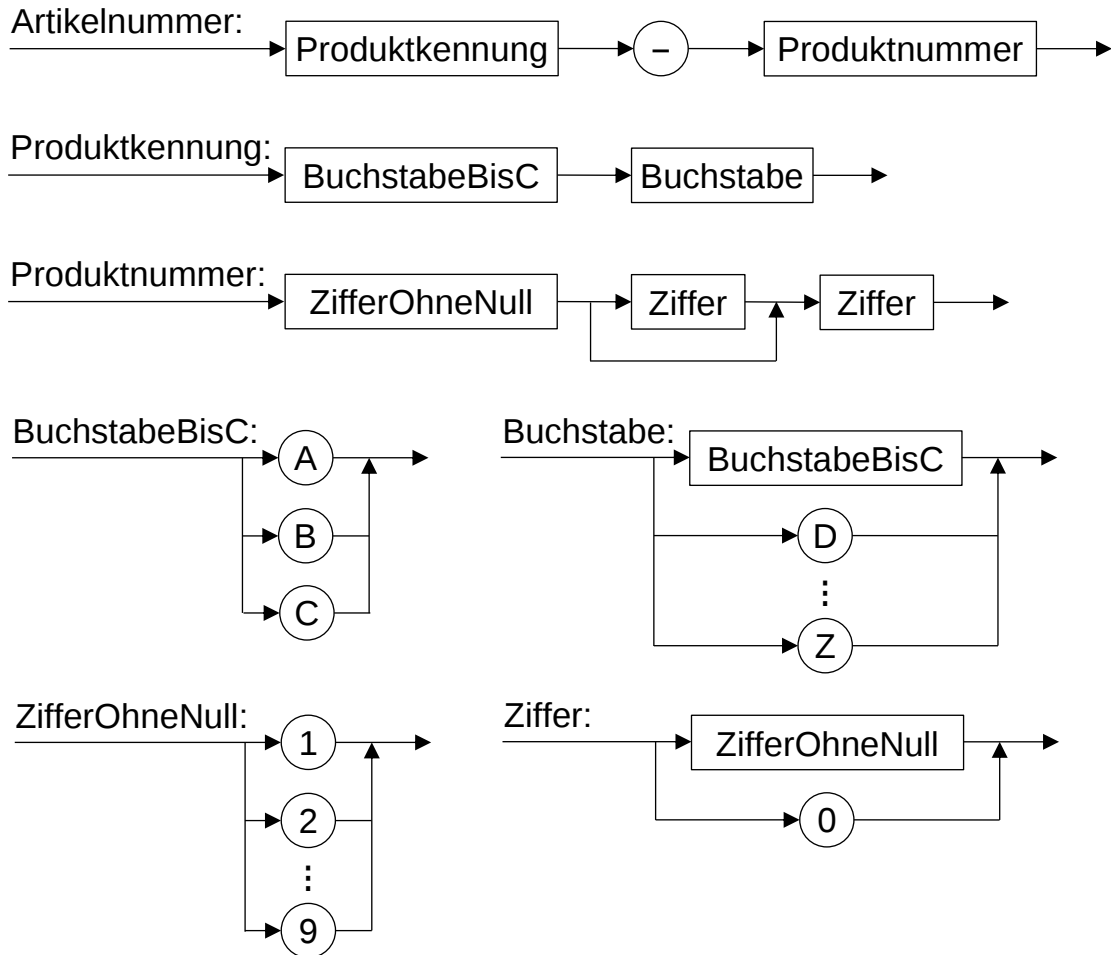
Beschränken Sie sich bei der Implementierung der Zustandsübergänge exemplarisch auf einen Zustand, von dem aus Übergänge zu mindestens zwei unterschiedlichen Zuständen möglich sind.

Sie dürfen dabei folgende Methoden der Klasse ZEICHENKETTE verwenden:

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

(Fortsetzung nächste Seite)

2. Im Onlineshop hat jeder Artikel eine Artikelnummer. Die Menge aller Artikelnummern bildet die formale Sprache L_A . Das abgebildete Syntaxdiagramm stellt deren Produktionsregeln dar.



- a) Geben Sie eine syntaktisch korrekte Artikelnummer an und formulieren Sie die zur Sprache L_A zugehörige Grammatik. Stellen Sie dabei die Produktionsregeln in formaler Textnotation dar (z. B. in EBNF).
- b) Begründen Sie, dass ein endlicher Automat, der genau die Wörter der Sprache L_A erkennt, mindestens zwei Endzustände haben muss.

(Fortsetzung nächste Seite)

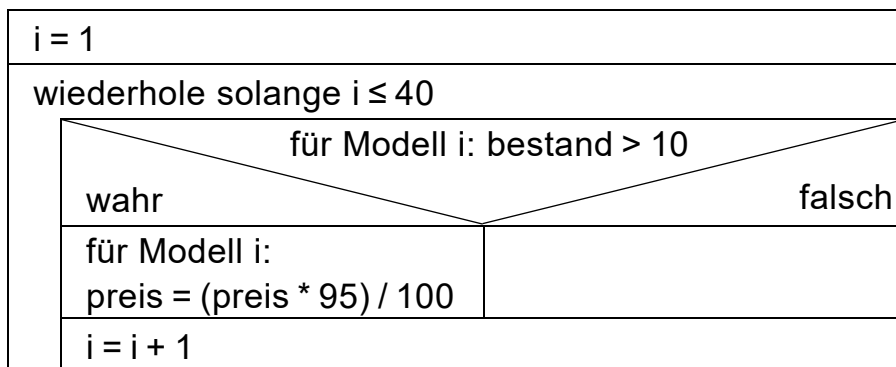
3. Im Onlineshop können mehrere Kunden gleichzeitig Artikel anschauen und bestellen.

2 a) Auf der Startseite werden mithilfe einer Methode *angeboteZeigen()* die aktuellen Sonderangebote angezeigt. Der Webserver bedient die eingehenden Anfragen nebenläufig. Beurteilen Sie, ob es durch das gleichzeitige Ausführen der Methode *angeboteZeigen()* zu Synchronisationsproblemen kommen kann.

3 b) Anna und Michael wählen jeweils das Buch „Java für Radler“ aus den angezeigten Sonderangeboten aus. Beiden wird der Artikel als verfügbar angezeigt und sie versuchen anschließend, das Buch zu bestellen. Annas Bestellung ist erfolgreich, Michael erhält dagegen eine Fehlermeldung, dass dieses Buch ausverkauft ist.

Erklären Sie, wie es zu dieser für Michael widersprüchlichen Fehlermeldung kam, und erläutern Sie, wie man dieses Problem verhindern kann.

8 4. Das Versandhaus hat 40 verschiedene Fahrradhelmmodelle im Sortiment. Deren Preise werden abhängig vom Bestand automatisiert angepasst. Dies wird mithilfe des folgenden Algorithmus realisiert:



(Fortsetzung nächste Seite)

Es steht eine Registermaschine mit folgendem Befehlssatz zur Verfügung:

load x	kopiert den Wert aus der Speicherzelle x in den Akkumulator
load (x)	kopiert den Wert aus derjenigen Speicherzelle in den Akkumulator, deren Adresse in der Speicherzelle x steht
loadi n	lädt die ganze Zahl n in den Akkumulator
store x	kopiert den Wert aus dem Akkumulator in die Speicherzelle x
store (x)	kopiert den Wert aus dem Akkumulator in diejenige Speicherzelle, deren Adresse in der Speicherzelle x steht
addi n	addiert die ganze Zahl n zum Wert im Akkumulator
subi n	subtrahiert die ganze Zahl n vom Wert im Akkumulator
muli n	multipliziert den Wert im Akkumulator mit der ganzen Zahl n
divi n	dividiert den Wert im Akkumulator durch die ganze Zahl n (ganzzahlige Division)
jmp x	springt zum Befehl in Speicherzelle x
jeq x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
jge x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator positiv oder gleich 0 ist
jle x	springt zum Befehl in Speicherzelle x, falls der Wert im Akkumulator negativ oder gleich 0 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
hold	beendet die Abarbeitung des Programms

Hinweis: Durch die Angabe einer Adresse in Klammern kann bei den Befehlen *load* und *store* indirekt adressiert werden. Enthält beispielsweise die Speicherzelle 50 den Wert 51 und die Speicherzelle 51 den Wert 3, so interpretiert der Befehl „*load (50)*“ den Wert 51 als Adresse einer Speicherzelle und lädt den Wert der Speicherzelle 51, also 3, in den Akkumulator.

Implementieren Sie den oben dargestellten Algorithmus für die gegebene Registermaschine. Gehen Sie davon aus, dass die Preise der Helmmodelle 1 bis 40 (in Cent) in den Speicherzellen 101 bis 140 und die zugehörigen Bestände entsprechend in den Speicherzellen 201 bis 240 gespeichert sind.

(Fortsetzung nächste Seite)

BE	
3	5. Nachdem die Artikel bestellt wurden, werden die Mitarbeiter im Lager aktiv. Um die obersten Regalfächer zu erreichen, benötigen sie eine lange Leiter. Wird diese benutzt, muss ein Helm getragen werden. Da in diesen Regalfächern nur Artikel gelagert werden, die selten bestellt werden, gibt es im Lager nur eine Leiter und einen Helm. Erläutern Sie, wie es in diesem Szenario zu einer Verklemmung kommen kann. Beschreiben Sie, unter welchen Umständen durch die Anschaffung einer weiteren Leiter die Verklemmung verhindert werden kann. 6. In der Versandabteilung steht den dortigen Mitarbeitenden zum Abschluss des Bestellvorgangs und Ausdruck der Versandetiketten ein Computer zur Verfügung, an dem sie sich mit einem Benutzernamen und einem Passwort anmelden. Das Passwort besteht aus einem Großbuchstaben gefolgt von drei Ziffern. Ein Mitarbeiter hat sein Passwort vergessen und versucht nun verschiedene Kombinationen, um sich wieder anmelden zu können. Jeder Versuch einschließlich der dazugehörigen Überprüfung der Anmeldedaten dauert 0,5 s. 2 a) Berechnen Sie, wie lange es höchstens dauert, bis der Mitarbeiter wieder angemeldet ist. 3 b) Beschreiben Sie drei Möglichkeiten, um den Computer vor unerlaubtem Zugriff besser zu schützen.
40	