

Abiturprüfung 2020

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 Kino „Altstadtklotz“ benötigt ein Verwaltungssystem und gibt dazu eine Software in Auftrag.

1. Die Software soll die vorhandenen Kinosäle und die aktuell laufenden Filme verwalten. Zur Planung des aktuellen Kinoprogramms muss die Möglichkeit bestehen, Spieltermine anzulegen. Dabei sind jedem erstellten Spieltermin genau ein Kinosaal und genau ein Film zugeordnet. Laufen also zeitgleich mehrere Filme, muss für jeden ein eigener Spieltermin existieren. Registrierte Kunden können online Reservierungen für einen oder mehrere Sitzplätze zu einem gewünschten Spieltermin vornehmen.

11 a) Modellieren Sie die beschriebene Situation in einem Klassendiagramm. Verwenden Sie dabei unter anderem die Klassen KINO und SPIELTERMIN. Führen Sie im Klassendiagramm mindestens drei Methoden an, die keine Standardmethoden zum Lesen oder Setzen eines Attributwerts sind.

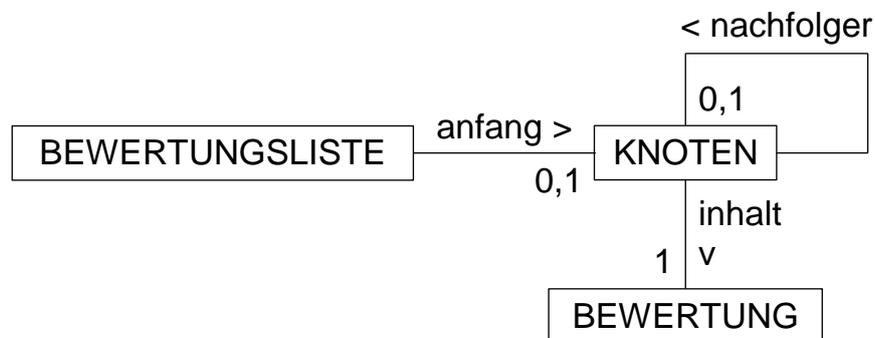
2 b) Das mit der Entwicklung der Software beauftragte Unternehmen muss eine Entscheidung treffen, wie Spieltermine und Kinosäle gespeichert werden. Als Datenstrukturen stehen dafür ein Feld und eine einfach verkettete Liste zur Auswahl. Erläutern Sie jeweils anhand eines Aspekts, wie Sie sich in der oben beschriebenen Situation entscheiden würden.

(Fortsetzung nächste Seite)

c) Wenn ein registrierter Kunde Karten online reservieren will und auf der Startseite korrekte Anmeldedaten eingegeben hat, dann erscheint eine Übersicht aller Spieltermine. Wählt der Benutzer einen Spieltermin aus, so wird der Saalplan angezeigt. Nach Angabe der gewünschten Plätze kann man auf „Weiter“ klicken. Wenn der Reservierungswunsch tatsächlich erfüllt werden kann, wird der Benutzer gefragt, ob er weitere Reservierungen vornehmen möchte. Bejaht er dies, so gelangt er wieder zur Übersicht der Spieltermine. Drückt er auf „Nein“, so erscheint eine Aufstellung seiner Reservierungen. Bestätigt er diese, wird alles gespeichert und er wird wieder zur Übersicht der Spieltermine geleitet. Alternativ kann er hier auch abrechnen, wodurch die Daten verworfen werden und ebenfalls erneut die Übersicht der Spieltermine angezeigt wird.

Stellen Sie die beschriebenen Abläufe in einem Zustandsübergangsdia-gramm dar.

2. Nach einem Jahr Produktivbetrieb will „Altstadtklotz“ seine Software um ein Bewertungssystem erweitern lassen. Kunden sollen zu einem Film, den sie gesehen haben, einen kurzen Bewertungstext schreiben und einen bis fünf Bewertungssterne vergeben können. Es wird vorgeschlagen, zur Verwaltung der Bewertungen in einer gemeinsamen Liste für alle Filme folgendes Klassendiagramm zugrunde zu legen:



a) Die Entwickler entschieden sich, die Bewertungseinträge nicht direkt, sondern über eine zusätzliche Klasse KNOTEN zu verketteten. Nennen Sie das zugrunde liegende Konzept sowie zwei damit verbundene Vorteile.

(Fortsetzung nächste Seite)

- b) Das Modell der Bewertungsliste soll nach einer Diskussion im Projektteam zusätzlich noch das Entwurfsmuster Kompositum verwenden. Begründen Sie, weshalb sich das Team für den Einsatz dieses Entwurfsmusters entschieden haben könnte.

Zeichnen Sie außerdem ein daran angepasstes Klassendiagramm, wobei auf die Nennung von Attributen und Methoden verzichtet werden darf.

- c) Im Folgenden soll die verbesserte Listenstruktur aus Teilaufgabe 2b zugrunde gelegt werden. Notieren Sie mithilfe einer auf dem Deckblatt angegebenen Programmiersprache eine Implementierung aller Klassen der Listenstruktur. Beschränken Sie sich bei den Methoden auf die folgenden der Klasse BEWERTUNGSLISTE und diejenigen, die zur Implementierung dieser Methoden erforderlich sind:

- *ein fuegen(bewertungNeu)* fügt ein Bewertungsobjekt vorne in die Bewertungsliste ein,
- *ausgeben(sterneMin, sterneMax)* gibt die Informationen derjenigen Bewertungen am Bildschirm aus, deren Anzahl von Sternen den angegebenen Bereich weder unter- noch überschreitet,
- *durchschnittGeben(filmtitel)* berechnet und liefert die durchschnittliche Anzahl von Sternen für einen bestimmten Film.

Wenden Sie so weit wie möglich das Prinzip der Rekursion an.

In der Klasse BEWERTUNGSLISTE darf die Methode *anzahlGeben(filmtitel)*, die die Anzahl aller Bewertungen zum übergebenen Filmtitel zurückgibt, als implementiert vorausgesetzt und verwendet werden.

Gehen Sie zudem davon aus, dass die Klasse BEWERTUNG bereits vollständig implementiert ist und folgende Methoden zur Verfügung stehen:

- *ausgeben()* gibt Informationen über die Bewertung am Bildschirm aus,
- *anzahlSterneGeben()* liefert die Zahl der Bewertungssterne,
- *gehörtZuFilm(filmtitel)* liefert genau dann *wahr*, wenn die Bewertung zum Film mit dem übergebenen Titel gehört.

(Fortsetzung nächste Seite)

3. Die Kinosoftware verwaltet die Kundendaten in einem geordneten Binärbaum, wobei als Schlüssel die Kundennummer dient, die sich aus den zwei Anfangsbuchstaben sowohl des Vor- als auch des Nachnamens sowie einer fortlaufenden Zahl zusammensetzt. Die Sortierung erfolgt lexikographisch.

4

a) Die Kunden mit den nachfolgenden Kundennummern sollen in einen zunächst leeren Binärbaum so eingefügt werden, dass der entstehende Binärbaum möglichst wenig Ebenen besitzt:

AnGe220, RuMo90, MaKi83, TeBo11, LuGi100, MaBr29, Schi1, BaLo22.

Zeichnen Sie einen solchen Baum und geben Sie zudem eine dazu passende Einfügereihenfolge an.

4

b) Gehen Sie nun davon aus, dass der Binärbaum aus neun Ebenen besteht und 295 Kunden enthält. Berechnen Sie unter der Bedingung, dass die Anzahl der Ebenen gleich bleiben soll, wie viele Kunden darin im besten Fall noch Platz finden.

Geben Sie zusätzlich für die oben beschriebene Bedingung einen Term an, der in Abhängigkeit von der Ebenenanzahl n und der Anzahl k der gespeicherten Kunden die Anzahl der freien Plätze beschreibt.

5

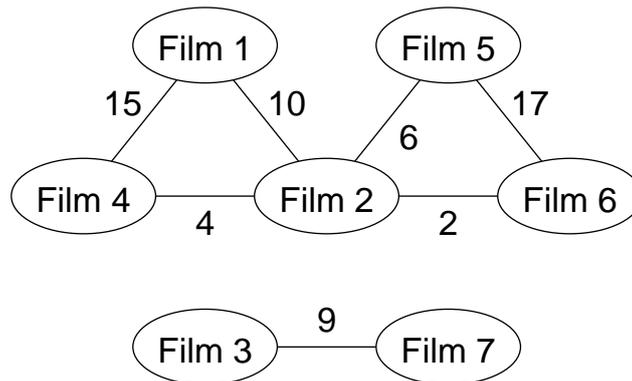
c) Bei der Implementierung des geordneten Binärbaums kommt das Entwurfsmuster Kompositum zum Einsatz.

Beschreiben Sie in Worten den rekursiven Algorithmus, den ein Knoten zum Einfügen eines neuen Kunden ausführt. Gehen Sie zudem kurz auf die Rolle des Abschlusses ein.

(Fortsetzung nächste Seite)

4. Kunden soll eine personalisierte Filmempfehlung gegeben werden. Dazu werden die aktuell laufenden Filme in einem gewichteten Graphen angeordnet. Die Kantengewichte geben an, von wie vielen Kinobesuchern beide Filme gesehen wurden. Keine Kante zwischen zwei Knoten bedeutet, dass diese Filmkombination noch nie von einem Kunden angesehen wurde.

Für die aktuell laufenden Filme ist folgender Graph entstanden:



- 5 a) Erstellen Sie die zugehörige Adjazenzmatrix. Geben Sie begründet an, welche Werte in der Matrix an den Stellen gespeichert sein sollen, für die kein Kantengewicht festgelegt ist. Gehen Sie dabei sowohl auf Einträge auf als auch außerhalb der Diagonalen der Matrix ein.
- 4 b) Zum gegebenen Graphen kommen folgende Daten hinzu:
- Drei Kunden, die bisher nur Film 1 und Film 2 gesehen haben, schauen sich nun Film 5 an.
 - Ein Kunde, der bisher nur Film 7 gesehen hat, schaut sich nun Film 2 und Film 3 an.

Zeichnen Sie den dadurch entstehenden Graphen.

(Fortsetzung nächste Seite)

Um den Graphen zu implementieren, speichert die Klasse GRAPH die Knoten in einem Feld *knoten*, die aktuelle Knotenanzahl im Attribut *anzahlKnoten* und die Adjazenzmatrix in einem zweidimensionalen Feld *matrix* ab.

Die Methode *besteEmpfehlungSuchen(i)* der Klasse GRAPH soll für den Knoten mit Index *i* denjenigen unter seinen Nachbarknoten ausgeben, zu dem die Kante mit dem größten Gewicht führt. Haben mehrere Kanten das gleiche Gewicht, soll der zuerst gefundene Knoten ausgegeben werden. Hat der Knoten keine Nachbarn, soll nichts ausgegeben werden.

8

c) Formulieren Sie einen Algorithmus (z. B. in Pseudocode) für diese Methode.

2

d) Nennen Sie zwei Gründe, weshalb die Ausgabe der Methode *besteEmpfehlungSuchen*, aufgerufen mit dem Index des zuletzt gesehenen Films, möglicherweise von den Kunden nicht als beste Empfehlung empfunden wird.

BE

Ein Spielwarengeschäft plant, als weitere Einnahmequelle einen Spieleverleih zu eröffnen. Für diesen soll eine Software entwickelt werden, die auch einen passwortgeschützten Onlinezugang für Kunden ermöglicht.

- 4
1. Nennen und beschreiben Sie knapp mindestens vier typische Phasen in der Softwareentwicklung.
 2. Die Software des Spieleverleihs soll den vorhandenen Spielebestand und die Kundendaten verwalten. Von jedem Spiel wird der Name und der Neupreis gespeichert. Der Spieleverleih hat in der Regel mehrere Exemplare eines Spiels. Für jedes Exemplar wird eine eindeutige Nummer sowie der Abnutzungsgrad als ganze Zahl auf einer Skala von 1 (neuwertig) bis 5 (mangelhaft) erfasst. Ein Ausleihvorgang bezieht sich stets auf genau ein Spielexemplar. Zu jedem Ausleihvorgang wird das Ausleihdatum festgehalten, die Zahl der Ausleihvorgänge pro Kunde ist nicht begrenzt.

- 6
- a) Erstellen Sie ein zu der beschriebenen Situation passendes Klassendiagramm mit den Klassen SPIELEVERLEIH, SPIEL, EXEMPLAR, AUSLEIHVORGANG und KUNDE. Auf die Angabe von Methoden und nicht im Text genannten Attributen kann verzichtet werden.

Die aktuellen Ausleihvorgänge eines Kunden werden in einer einfach verketteten Liste verwaltet, die auf dem Entwurfsmuster Kompositum basiert und die Trennung von Struktur und Daten berücksichtigt.

- 4
- b) Zeichnen Sie das zugrunde liegende Klassendiagramm. Auf die Angabe von Attributen und Methoden kann verzichtet werden.

Gehen Sie im Folgenden davon aus, dass jeder Ausleihvorgang eine Referenz auf das betreffende Exemplar und jedes Exemplar eine Referenz auf das zugehörige Spiel speichert.

Die Kundin Johanna Huber hat sich am 03.04.2020 zwei Exemplare (Abnutzungsgrade: 1 und 4) des Spiels „Boccia“ (Neupreis: 35 €) ausgeliehen, wofür entsprechende Ausleihvorgänge in die zunächst leere Ausleihliste *liste1* dieser Kundin eingefügt wurden.

(Fortsetzung nächste Seite)

7

c) Zeichnen Sie das zugehörige Objektdiagramm. Beschränken Sie sich bei den Attributen auf diejenigen, zu denen im Text ein Wert angegeben ist.

5

d) Johanna Huber hat das stärker abgenutzte Boccia-Spiel sieben Tage nach Ablauf der Ausleihfrist zurückgegeben. Für die Überziehung ist eine Gebühr fällig, deren (ungerundete) Höhe in der Methode *gebuehrBerechnen* der Klasse AUSLEIHVORGANG nach folgender Formel ermittelt wird:

$$\text{gebuehr} = \text{restwert} * \text{anzahlÜberzogeneTage} / 28$$

Dabei wird der Restwert eines Spielexemplars folgendermaßen berechnet:

$$\text{restwert} = \text{neupreis} * (6 - \text{abnutzungsgrad}) / 5$$

Stellen Sie die Berechnung der Gebühr für das zu spät zurückgegebene Spiel in einem Sequenzdiagramm dar. Gehen Sie davon aus, dass in der Klasse EXEMPLAR die Methode *restwertBerechnen* und in der Klasse SPIEL die Methode *neupreisGeben* zur Verfügung stehen.

7

e) Gibt ein Kunde ein Spielexemplar zurück, soll der entsprechende Ausleihvorgang aus der Liste entfernt werden.

Die Klasse AUSLEIHLISTE soll dazu die Methode *entfernen(exemplar)* besitzen.

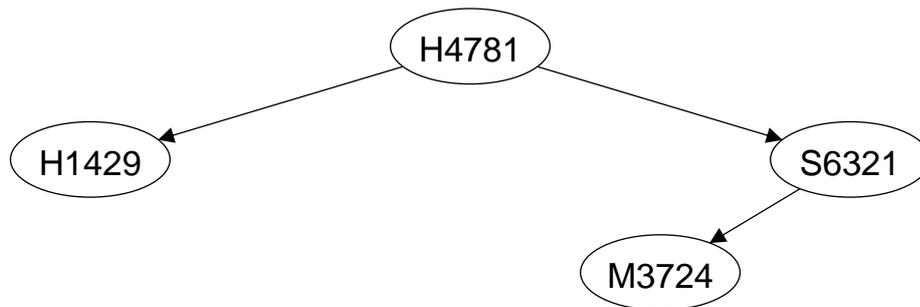
Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *entfernen* in der Klasse AUSLEIHLISTE und aller dafür benötigten Methoden in den Klassen der Listenstruktur. Verwenden Sie dabei das Prinzip der Rekursion.

Die Methode *exemplarGeben()* der Klasse AUSLEIHVORGANG, die eine Referenz auf das zugehörige Exemplar zurückgibt, darf als bereits implementiert vorausgesetzt werden.

(Fortsetzung nächste Seite)

3. Die Kundendaten sollen in einem geordneten Binärbaum verwaltet werden. Die Daten werden lexikographisch nach den eindeutigen Kundennummern sortiert. Die Kundennummer wird aus dem Anfangsbuchstaben des Nachnamens sowie einer vierstelligen positiven Zahl gebildet.

- 5 a) Gegeben ist folgender geordneter Binärbaum:



Geben Sie jede mögliche Reihenfolge an, in der die Kundennummern in einen anfangs leeren Baum eingefügt worden sein können, sodass der gegebene Baum entsteht.

Zeichnen Sie außerdem den Baum, der entsteht, wenn in den gegebenen Baum nun nacheinander die Kundennummern R7491, N2855 sowie L7762 eingefügt werden.

- 4 b) Nennen und beschreiben Sie ein Verfahren, um alle Kunden nach Kundennummern sortiert auszugeben. Begründen Sie kurz, warum dieses Verfahren in der Regel keine nach den Nachnamen der Kunden sortierte Ausgabe erzeugt.

- 2 c) Der Spieleverleih rechnet in der Anfangsphase mit 2500 Kunden. Bestimmen Sie die Anzahl der Ebenen, die ein dazugehöriger geordneter Binärbaum mindestens haben muss.

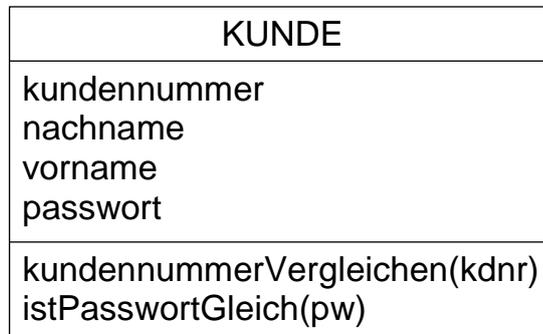
- 3 d) Nennen Sie je einen Vor- und einen Nachteil der Implementierung von Maßnahmen, die sicherstellen, dass die Ebenenanzahl des Binärbaums stets minimal ist.

Bei der Implementierung des geordneten Binärbaums zur Verwaltung der Kundendaten mithilfe einer Klasse KUNDENBAUM kommen das Entwurfsmuster Kompositum sowie das Konzept der Trennung von Struktur und Daten zum Einsatz.

(Fortsetzung nächste Seite)

e) Geben Sie an, worin sich das zugehörige Klassendiagramm von Ihrem aus Teilaufgabe 2b im Wesentlichen unterscheidet.

f) Die Klasse KUNDE ist folgendermaßen beschrieben:



Die Methode *kundennummerVergleichen(kdnr)*

- gibt 1 zurück, wenn der Parameter *kdnr* einen größeren Wert als das Attribut *kundennummer* des ausführenden Objekts hat,
- gibt 0 zurück, wenn der Parameter *kdnr* den gleichen Wert wie das Attribut *kundennummer* des ausführenden Objekts hat,
- gibt -1 zurück, wenn der Parameter *kdnr* einen kleineren Wert als das Attribut *kundennummer* des ausführenden Objekts hat.

Die Methode *istPasswortGleich(pw)* gibt genau dann *wahr* zurück, wenn der Wert des Parameters *pw* mit dem des Attributs *passwort* des ausführenden Objekts übereinstimmt.

Um sich online beim Spieleverleih anzumelden, muss ein Kunde seine Kundennummer und sein Passwort eingeben.

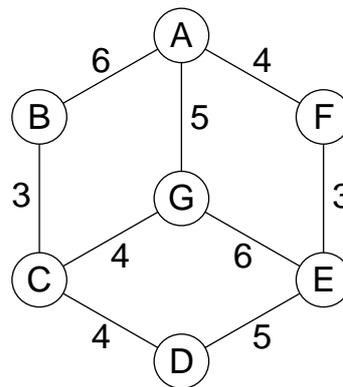
Die Klasse KUNDENBAUM soll eine Methode *passwortPruefen(kdnr, pw)* bereitstellen, die genau dann *wahr* zurückgibt, wenn in der Baumstruktur ein Kunde mit Kundennummer *kdnr* und Passwort *pw* eingetragen ist.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung dieser Methode für die Klasse KUNDENBAUM und der dafür nötigen Methoden aller weiteren Klassen der Baumstruktur. Nutzen Sie dabei das Ordnungsprinzip des Binärbaums. Die Klasse KUNDE kann als vollständig implementiert vorausgesetzt werden.

(Fortsetzung nächste Seite)

4. Das Spielwarengeschäft hat für den Spieleverleih ein neues Brettspiel angeschafft. Bei diesem Brettspiel hat jeder Spieler eine Spielfigur. Auf dem Spielbrett sind Felder eingezeichnet, auf denen eine oder auch mehrere Spielfiguren stehen können. Manche Felder sind mit Linien verbunden, die jeweils mit einer positiven ganzen Zahl beschriftet sind.

Das Spielbrett kann durch einen Graphen modelliert werden, bei dem die Felder durch die Knoten und die Verbindungslinien durch die Kanten repräsentiert werden. Die folgende Abbildung zeigt einen solchen Graphen für einen Ausschnitt des Spielbretts:



6

- a) Erstellen Sie die Adjazenzmatrix des abgebildeten Graphen und geben Sie zwei grundlegende Eigenschaften dieses Graphen an.

(Fortsetzung nächste Seite)

Wenn ein Spieler an der Reihe ist, würfelt er zunächst mit zwei Würfeln. Die Summe der gewürfelten Augenzahlen erhält der Spieler als Punkte auf sein Punktekonto gutgeschrieben. Anschließend kann er sich entscheiden, ob er seine Spielfigur bewegen möchte. Die Spielfigur kann von ihrem Feld aus nur auf diejenigen Felder bewegt werden, zu denen eine Verbindungslinie führt. Ein Zug kann über mehrere Verbindungslinien erfolgen, jedoch darf in einem Zug kein Feld zweimal besucht werden. Bei jedem Wechsel zu einem Nachbarfeld wird dem Spieler der Wert der betreffenden Verbindungslinie von seinem Punktekonto abgezogen. Reicht sein Punktestand nicht aus, kann seine Spielfigur die Verbindungslinie nicht nutzen.

5 b) Ermitteln Sie für den oben abgebildeten Ausschnitt des Spielbretts alle Züge, die eine Spielfigur, die auf dem Feld A steht, ausführen kann, wenn deren Spieler 12 Punkte auf seinem Punktekonto gesammelt hat. Geben Sie dabei auch an, wie viele Punkte bei den einzelnen Zügen jeweils übrig bleiben.

10 c) Ein elektronischer Assistent zeigt bei gegebener Position der Spielfigur und gegebenem Punktestand alle möglichen Züge sowie die verbleibenden Punkte an.

Formulieren Sie einen Algorithmus (z. B. in Pseudocode), der die beschriebene Aufgabe des elektronischen Assistenten umsetzt.

Gehen Sie davon aus, dass die Klasse GRAPH die Knoten in einem Feld *knoten*, die aktuelle Knotenanzahl im Attribut *anzahlKnoten* und die Adjazenzmatrix in einem zweidimensionalen Feld *matrix* abspeichert.

BE

Das Gymnasium in Turingen ist auf einem weitläufigen Campus angesiedelt.

1. Auf dem Schulcampus gibt es drei Gebäude, deren Räume alle eine eindeutige Raumbezeichnung haben. Diese besteht aus einem Buchstaben für das Gebäude (A, B oder C), gefolgt von einem Bindestrich und einer zwei- oder dreistelligen Raumnummer (01 bis 99 bzw. 001 bis 999). Beispielsweise sind A-08 und C-239 gültige Raumbezeichnungen. Die Menge aller gültigen Raumbezeichnungen bildet die formale Sprache L.

5

a) Gegeben ist die Grammatik G mit dem Alphabet $\Sigma = \{A, B, C, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, der Menge der Nichtterminale $V = \{\langle \text{Raum} \rangle, \langle \text{Gebäude} \rangle, \langle \text{Nummer} \rangle, \langle Z \rangle, \langle Z_n N \rangle\}$, dem Startsymbol $\langle \text{Raum} \rangle$ und den folgenden Produktionsregeln:

(R1) $\langle \text{Raum} \rangle \rightarrow \langle \text{Gebäude} \rangle - \langle \text{Nummer} \rangle$

(R2) $\langle \text{Gebäude} \rangle \rightarrow A \mid B \mid C$

(R3) $\langle \text{Nummer} \rangle \rightarrow \langle Z_n N \rangle \langle Z \rangle \mid \langle Z \rangle \langle Z_n N \rangle \mid \langle Z \rangle \langle \text{Nummer} \rangle$

(R4) $\langle Z \rangle \rightarrow 0 \mid \langle Z_n N \rangle$

(R5) $\langle Z_n N \rangle \rightarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Die von G erzeugte Sprache stimmt nicht mit L überein. Geben Sie einerseits die Ableitung eines Wortes an, das durch G beschrieben wird, aber nicht zu L gehört. Nennen Sie andererseits ein Wort aus L, das nicht von G erzeugt wird.

6

b) Geben Sie das Zustandsübergangdiagramm eines endlichen Automaten an, der genau die Raumbezeichnungen der Sprache L akzeptiert.

2

c) Grenzen Sie im Sachzusammenhang die Begriffe Semantik und Syntax voneinander ab.

(Fortsetzung nächste Seite)

2. Auf dem Campus stehen fünf Tabletkoffer zur Verfügung, die von Lehrkräften für einzelne Stunden in einem Online-Buchungssystem ausgewählt und reserviert werden können. Für den Medienraum, in dem die Tabletkoffer aufbewahrt werden, sind beim Hausmeister Herrn Immerda zwei Schlüssel hinterlegt.

4 a) Auf Anfrage zeigt das Buchungssystem für eine bestimmte Stunde die noch freien Koffer an. In einem zweiten Schritt nimmt es die Reservierung für den ausgewählten Koffer entgegen. Das Buchungssystem prüft dabei erneut, ob der Koffer noch frei ist, nimmt dann ggf. die Buchung vor und bestätigt diese. Reservierungen können nebenläufig getätigt werden. Hierbei kann es vorkommen, dass mehrere Lehrkräfte für denselben Koffer und dieselbe Stunde eine Reservierungsbestätigung erhalten. Erklären Sie anhand eines konkreten Szenarios das Zustandekommen dieses Fehlers.

3 b) Geben Sie den kritischen Abschnitt des in Teilaufgabe 2a beschriebenen Ablaufs an. Nennen und beschreiben Sie allgemein ein Konzept, mit dem derartige Mehrfachreservierungen verhindert werden können.

5 c) Die Lehrkräfte durchlaufen beim Abholen und Zurückbringen der Koffer der Reihe nach folgende Schritte:

- (S1) einen der zwei Schlüssel beim Hausmeister abholen (falls vorhanden, sonst warten)
- (S2) in den Medienraum gehen
- (S3) reservierten Koffer entnehmen (falls vorhanden, sonst warten) bzw. Koffer zurückstellen
- (S4) zum Hausmeister gehen
- (S5) Schlüssel beim Hausmeister abgeben

Zeigen Sie anhand eines Beispiels, dass es so zu einer Verklemmung kommen kann. Beschreiben Sie einen veränderten Ablauf, mit dem Verklemmungen verhindert werden können.

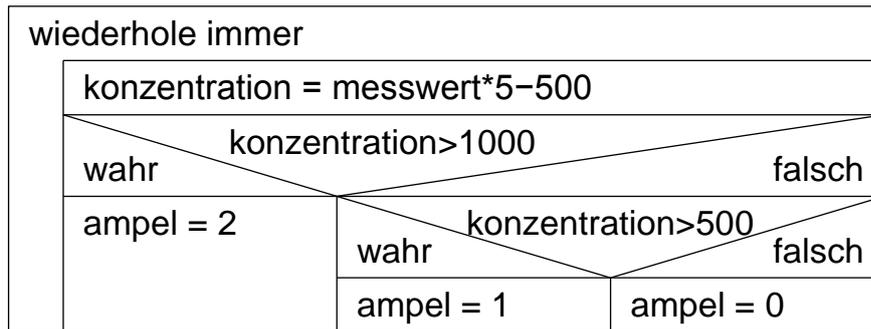
(Fortsetzung nächste Seite)

3. In allen Klassenzimmern auf dem Schulcampus befinden sich Sensoren, die den CO₂-Gehalt der Raumluft messen. Die Auswertung der Messdaten erfolgt durch 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
inc	erhöht den Wert im Akkumulator um 1
dec	verringert den Wert im Akkumulator um 1
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
jmp x	springt zum Befehl in Speicherzelle x
jeq 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
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 Programmabarbeitung

(Fortsetzung nächste Seite)

- a) Die Registermaschine steuert eine Warnampel, die je nach CO₂-Konzentration unterschiedlich leuchtet. Dazu verwendet sie den jeweils aktuellen Sensormesswert (eine natürliche Zahl größer oder gleich 100, die in Speicherzelle 202 hinterlegt ist), rechnet diesen gemäß den Angaben des Sensorherstellers in die CO₂-Konzentration um und legt das Ergebnis in Speicherzelle 203 sowie den neuen Farbcode der Ampel (0 für grün, 1 für gelb, 2 für rot) in Speicherzelle 201 ab. Das folgende Struktogramm beschreibt den Programmablauf:



Setzen Sie dieses Struktogramm in ein Programm für die gegebene Registermaschine um.

(Fortsetzung nächste Seite)

Eine CO₂-Messung durch den Sensor dauert deutlich länger als ein Durchlauf der Wiederholung aus Teilaufgabe 3a. Damit der in Speicherzelle 202 hinterlegte aktuelle Messwert nicht unnötig oft ausgelesen werden muss, soll die Dauer des Durchlaufs durch Hinzufügen einiger Programmzeilen verlängert werden. Hierzu ist in Speicherzelle 210 ein positiver ganzzahliger Startwert n hinterlegt.

```
        load 210
xx:    addi 2
        dec
        dec
        dec
        jgt xx
```

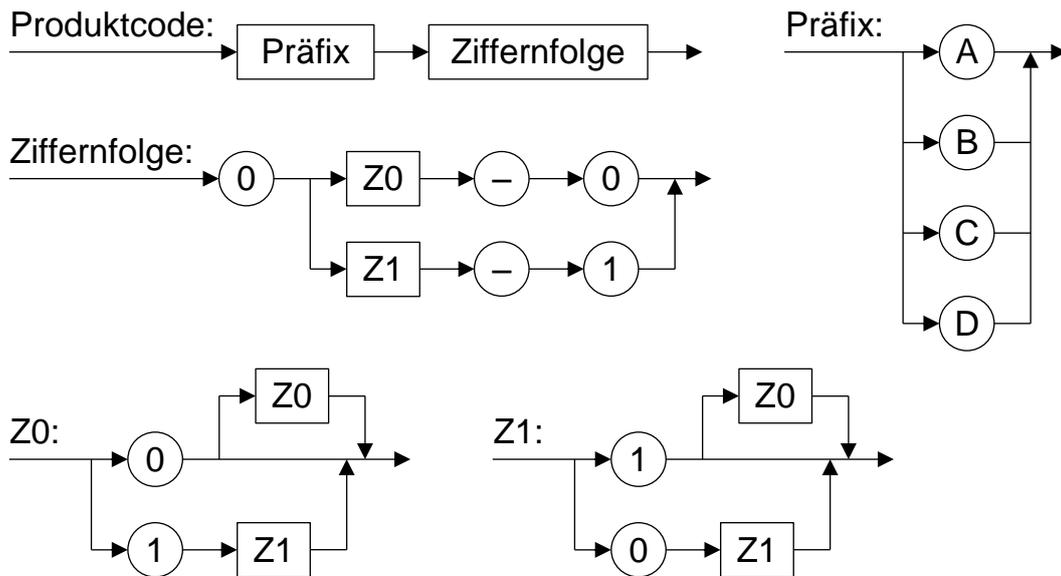
Dabei ist xx die Adresse der Speicherzelle, in der der Befehl `addi 2` steht.

- 4 b) Geben Sie für die Startwerte $n = 1, 2, 3, 10$ und 100 jeweils an, wie viele zusätzliche Befehlsausführungen der hinzugefügte Programmcode bewirkt. Geben Sie außerdem einen Term an, der die Anzahl dieser Befehlsausführungen in Abhängigkeit von n beschreibt.
- 3 c) Der gegebene Ausschnitt soll zu einer Verzögerung von etwa 2 s führen. Bestimmen Sie einen geeigneten Startwert n , wenn die Ausführung eines Befehls im Mittel 10^{-7} s dauert.

IV.

BE

1. Die Produkte eines Versandhandels sind intern mit Produktcodes versehen, die von Scannern automatisch gelesen werden können. Zulässige Produktcodes werden durch eine Grammatik G mit Startsymbol \langle Produktcode \rangle beschrieben. Die Produktionsregeln von G sind in folgendem Syntaxdiagramm dargestellt:



- 4 a) Nennen Sie das Alphabet von G und geben Sie die Produktionsregeln von G in einer formalen Textnotation (z. B. EBNF) an.
- 4 b) Von den Zeichenketten „C01011–0“ und „C01011–1“ ist genau eine ein syntaktisch korrekter Produktcode. Notieren Sie für diesen eine Ableitung. Geben Sie zudem an, welche Eigenschaft die Anzahl der Einsen in zulässigen Produktcodes allgemein hat.
- 5 c) Geben Sie das Zustandsübergangdiagramm eines endlichen Automaten an, der genau die nach dem oben angegebenen Syntaxdiagramm gebildeten Produktcodes akzeptiert.

(Fortsetzung nächste Seite)

2. Eine große Online-Videoplattform mit Millionen von Kunden betreibt weltweit eine Vielzahl von Servern, die Videos bereithalten. Kunden, die ein Video abrufen, werden an einen dieser Videosever vermittelt.

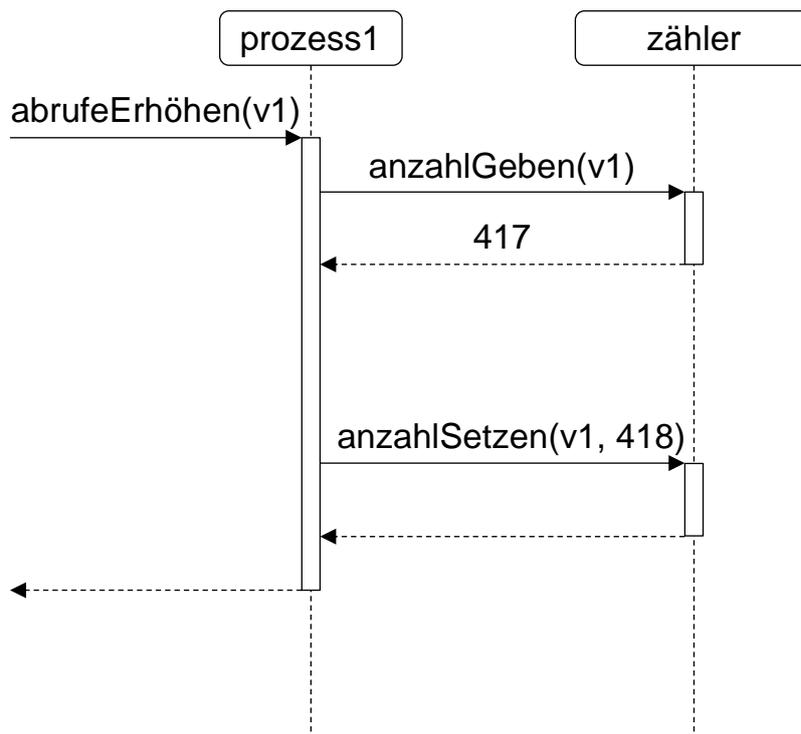
Auf einem Zentralserver läuft ein Auswertungsprogramm, das mitzählt, wie oft die einzelnen Videos jeweils abgerufen wurden. Dazu melden die Videosever jeden Abruf an diesen Zentralserver.

Bei jedem Abruf eines Videos wird auf dem Zentralserver von einem Prozess, der die Kommunikation mit dem Videosever regelt, folgende Methode ausgeführt:

```

Methode abrufeErhöhen(video)
    abrufe = zähler.anzahlGeben(video)
    abrufe = abrufe+1
    zähler.anzahlSetzen(video, abrufe)
endeMethode
  
```

Folgendes Sequenzdiagramm veranschaulicht beispielhaft den Ablauf dieser Methode:



4

- a) Zeichnen Sie ein weiteres Sequenzdiagramm, aus dem hervorgeht, wie es zu falschen Zählergebnissen kommen kann, wenn mehrere Prozesse die Methode *abrufeErhöhen* parallel ausführen.

(Fortsetzung nächste Seite)

5

b) Erläutern Sie, wie das Problem der falschen Zählergebnisse mithilfe einer geeigneten Synchronisationsmaßnahme beseitigt werden kann.

Beurteilen Sie zudem, ob es bei Anwendung dieser Synchronisationsmaßnahme zu einer Verklemmung kommen kann.

3

c) Das Verarbeiten einer Abrufmeldung dauert im Schnitt 10 Millisekunden. Bestimmen Sie die maximale Anzahl an Meldungen, die der Zentralserver pro Tag verarbeiten kann.

Jeder Videoserver kann die Meldung an den Zentralserver entweder nach jedem einzelnen Abruf oder als Sammelmeldung einmal pro Stunde durchführen. Geben Sie zu jeder der beiden Varianten einen Nachteil an.

(Fortsetzung nächste Seite)

3. 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
sub x	subtrahiert den Wert aus der Speicherzelle x vom Wert im Akkumulator
mul x	multipliziert den Wert im Akkumulator mit dem Wert in Speicherzelle x
div x	dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x (ganzzahlige Division)
mod x	dividiert den Wert im Akkumulator durch den Wert in Speicherzelle x und speichert den ganzzahligen Rest dieser Division im Akkumulator
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)
modi n	dividiert den Wert im Akkumulator durch die ganze Zahl n und speichert den ganzzahligen Rest dieser Division im Akkumulator
jmp x	springt zum Befehl in der Speicherzelle x
jmpn x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator negativ ist
jmpz x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
jmpp x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator positiv ist
jmpnn x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator nicht negativ ist
jmpnz x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator ungleich 0 ist
jmpnp x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator nicht positiv ist
hold	beendet die Abarbeitung des Programms

(Fortsetzung nächste Seite)

3 a) Gegeben ist das folgende Programm für die Registermaschine:

```
1: load 300
2: jmpnn 4
3: muli -1
4: store 301
5: hold
```

Stellen Sie den zugrunde liegenden Algorithmus graphisch dar (z. B. durch ein Struktogramm). Bezeichnen Sie dabei den in Speicherzelle 300 abgelegten Wert mit x und den berechneten Wert in Speicherzelle 301 mit y . Geben Sie zudem die Bedeutung von y an.

Folgender Algorithmus berechnet zu einer ganzzahligen Basis und einem nichtnegativen ganzzahligen Exponenten die Potenz:

```
Methode potenz(basis, exponent)
  wenn exponent gleich 0
    gib 1 zurück
  sonst
    wenn exponent gerade
      gib potenz(basis*basis, exponent/2) zurück
    sonst
      gib potenz(basis, exponent-1)*basis zurück
    endeWenn
  endeWenn
endeMethode
```

4 b) Stellen Sie die Berechnung der Potenz für den Methodenaufruf $\text{potenz}(2, 5)$ schrittweise dar.

Geben Sie zudem an, wie sich die Zahl der rekursiven Aufrufe verändert, wenn ein positiver Exponent verdoppelt wird.

8 c) Übertragen Sie diesen Algorithmus in ein Programm für eine Registermaschine mit dem oben genannten Befehlssatz. Machen Sie deutlich, in welchen Speicherzellen die Werte von Basis, Exponent und Ergebnis abgelegt werden.