



Teil 1: Beispielaufgaben

Die Illustrierenden Prüfungsaufgaben (Teil 1: Beispielaufgaben, Teil 2: Erläuterungen und Lösungsvorschläge) dienen der einmaligen exemplarischen Veranschaulichung von Struktur, Anspruch und Niveau der Abiturprüfung auf grundlegendem bzw. erhöhtem Anforderungsniveau im neunjährigen Gymnasium in Bayern.

Informatik

grundlegendes Anforderungsniveau

Arbeitszeit: 255 Minuten

Bei der Bearbeitung der Aufgaben dürfen folgende Hilfsmittel verwendet werden:

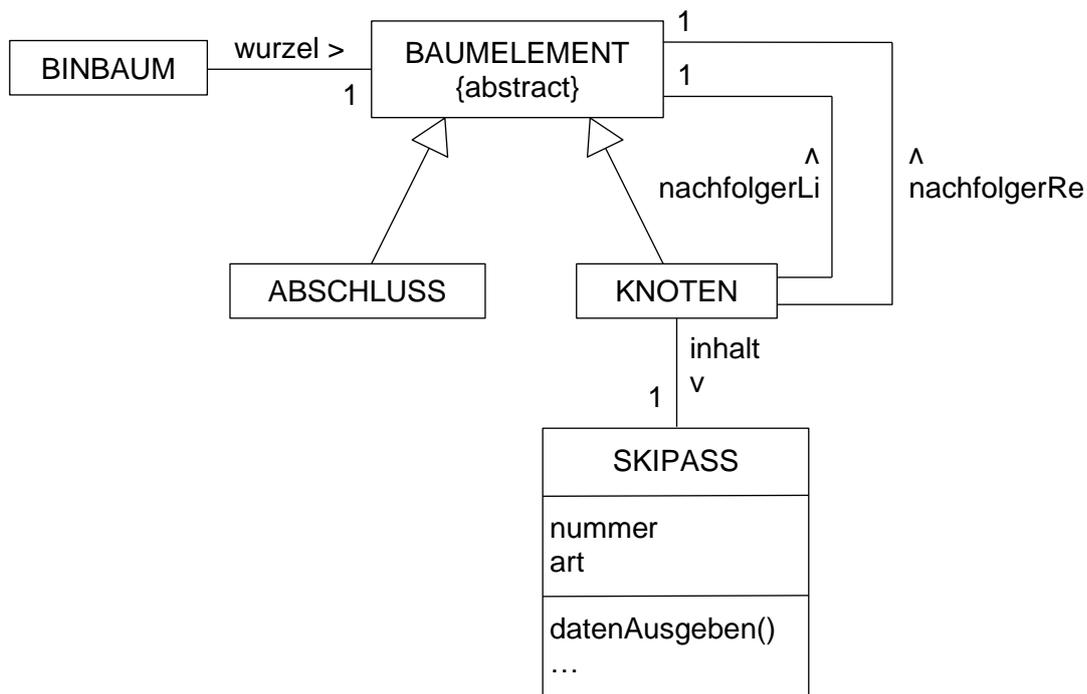
- das vom Staatsministerium genehmigte Dokument mit mathematischen Formeln
- ein Taschenrechner, der den vom Staatsministerium getroffenen Regelungen entspricht

Es werden vier Aufgaben zur Auswahl gestellt, von denen drei bearbeitet werden müssen.

Das Geheft mit den Aufgabenstellungen ist abzugeben.

Im Skigebiet Turingberg wird der Zugang zu den Skiliften über Flügeltüren gesteuert, die sich automatisch öffnen, wenn man einen gültigen Skipass an ein Lesegerät hält. Dabei wird die eindeutige Nummer des Skipasses drahtlos übertragen und als Eingabe an entsprechende Methoden eines Softwaresystems weitergegeben. Das Softwaresystem überprüft dann die Gültigkeit des entsprechenden Skipasses und veranlasst gegebenenfalls das Öffnen der Flügeltüren. Die Daten der aktuell gültigen Skipässe sind vom eingesetzten Softwaresystem als Objekte einer Klasse SKIPASS angelegt.

- 1 Um eine schnelle Suche zu ermöglichen, werden die Skipass-Objekte in einem geordneten Binärbaum gemäß folgendem Klassendiagramm verwaltet. Als Schlüssel dient die Skipassnummer, die eine positive ganze Zahl ist.



Als Werte von *art* sind ausschließlich die Buchstaben E, J und K vorgesehen, je nachdem, ob es sich um einen Skipass für Erwachsene, Jugendliche bzw. Kinder handelt.

- a Zeichnen Sie einen balancierten geordneten Binärbaum, in dem 15 Skipass-Objekte verwaltet werden. Wählen Sie als Schlüssel 15 zweistellige Skipassnummern selbst und verwenden Sie diese als Bezeichner der Knoten.

4

Begründen Sie, warum ein balancierter Baum für die Suche von Vorteil ist, und geben Sie eine Reihenfolge zum Einfügen Ihrer Skipassnummern an, die zu dem von Ihnen gezeichneten Baum führt.

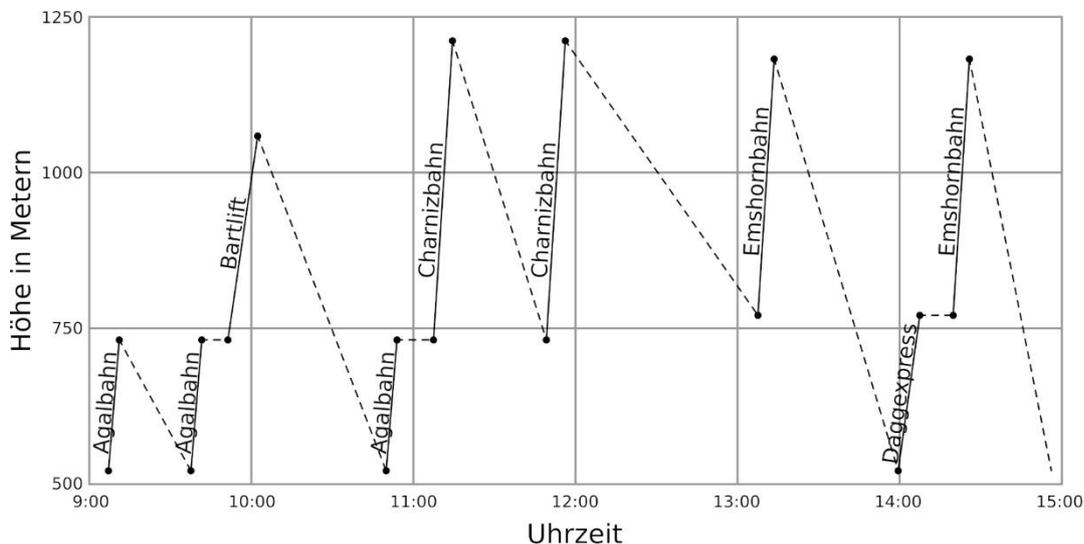
(Fortsetzung nächste Seite)

b Die Klasse BINBAUM soll folgende Methoden besitzen:

- *istVorhanden(skipassnr)* gibt genau dann *wahr* zurück, wenn zur übergebenen Skipassnummer ein entsprechendes Objekt im Binärbaum existiert.
- *sortiertAusgeben(skipassart)* gibt die Informationen von allen Skipässen einer bestimmten Art (z. B. Jugendliche) nach der Skipassnummer aufsteigend sortiert aus.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung dieser Methoden und aller dazu nötigen Methoden in der gegebenen Baumstruktur. Verwenden Sie dabei so weit wie möglich das Prinzip der Rekursion und nutzen Sie die Vorteile des geordneten Binärbaums. In der Klasse SKIPASS dürfen die Standardmethoden *nummerGeben* und *artGeben* sowie eine Methode *datenAusgeben* zur Ausgabe der Attributwerte (z. B. auf einem Bildschirm) als bereits implementiert vorausgesetzt werden.

2 Für jeden Skipass sollen alle damit erfolgten Liffahrten eines bestimmten Tages gespeichert werden. Interessierte Skifahrerinnen und Skifahrer sollen sich daraus ein Diagramm generieren lassen können, das beispielsweise wie folgt aussieht:

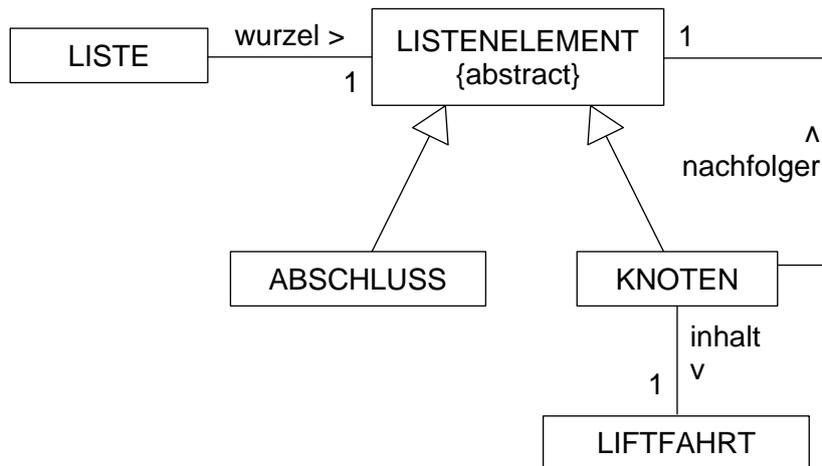


Die benötigten Daten der einzelnen Skilifte sind im System als Objekte einer Klasse SKILIFT bereits vorhanden, wobei die Dauer einer Liffahrt in (ganzen) Minuten gespeichert ist. Außerdem steht im System durch Aufruf der Methode *Uhr.minutenGeben()* die aktuelle Uhrzeit in ganzen Minuten seit Mitternacht zur Verfügung.

a Zeichnen Sie ein Klassendiagramm, das die Klassen SKIPASS, SKILIFT, LISTE und LIFTFAHRT enthält und auf dessen Grundlage die oben beschriebene Funktionalität implementiert werden kann. Geben Sie darin alle Attribute einschließlich geeigneter Datentypen an, die zur Erstellung eines Diagramms wie dem oben dargestellten nötig sind. Auf die Angabe von Methoden kann verzichtet werden. Berücksichtigen Sie, dass es im Skigebiet Turingberg auch Skipässe gibt, die an mehreren Tagen gültig sind.

(Fortsetzung nächste Seite)

Die für einen bestimmten Skipass zu speichernden Liftfahrten eines bestimmten Tages sollen in einer einfach verketteten Liste gemäß folgendem Klassendiagramm verwaltet werden.



b Erklären Sie, warum man als Datenstruktur eine einfach verkettete Liste gewählt haben könnte. Erläutern Sie außerdem, warum es sinnvoll ist, das Entwurfsmuster Kompositum anzuwenden, und stellen Sie dar, inwiefern bei der gegebenen Umsetzung die Trennung von Struktur und Daten berücksichtigt wurde.

3

c Unmittelbar nachdem der Zugang zu einem Skilift gewährt wurde, wird die Methode *liftfahrt-Speichern(skilift)* des dazugehörigen Skipass-Objekts aufgerufen, wobei eine Referenz auf das entsprechende Skilift-Objekt *skilift* übergeben wird. Das Skipass-Objekt ruft daraufhin die Methode *liftfahrtEinfuegen(skilift)* desjenigen zum Skipass-Objekt gehörenden Listenobjekts auf, das die Liftfahrten des aktuellen Tages mit den zugehörigen Uhrzeiten aufzeichnet.

4

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *liftfahrtEinfuegen* und aller dazu nötigen Methoden in der gegebenen Listenstruktur, wobei das neue Liftfahrt-Objekt am Anfang der Liste eingefügt werden soll, damit der Vorgang des Einfügens vergleichsweise schnell erfolgt.

d Zur Erstellung eines Diagramms werden die Daten der genutzten Skilifte in chronologischer Reihenfolge benötigt.

5

In der Klasse LISTE soll eine Methode *liftfahrtenAusgeben* zur Verfügung stehen, welche die Namen der gefahrenen Skilifte chronologisch (beginnend mit der ersten Liftfahrt des Tages) am Bildschirm ausgibt, wobei die Liftfahrten wie in Teilaufgabe 2c beschrieben in die Liste eingefügt wurden.

Notieren Sie in einer auf dem Deckblatt angegebenen Programmiersprache eine mögliche Implementierung der Methode *liftfahrtenAusgeben* in der Klasse LISTE und aller dafür benötigten Methoden in den Klassen der Listenstruktur. Verwenden Sie dabei das Prinzip der Rekursion. Die Klasse SKILIFT darf dabei als vollständig implementiert vorausgesetzt werden. Insbesondere stehen Methoden zum Geben und Setzen der Attributwerte zur Verfügung.

30

1 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)
jmp x	springt zum Befehl in der Speicherzelle x
jlt x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator negativ ist
jeq x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator gleich 0 ist
jgt x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator positiv ist
jle x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator negativ oder gleich 0 ist
jne x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator ungleich 0 ist
jge x	springt zum Befehl in der Speicherzelle x, falls der Wert im Akkumulator positiv oder gleich 0 ist
hold	beendet die Abarbeitung des Programms

a Beschreiben Sie einen möglichen Befehlszyklus einer Registermaschine am Beispiel des Maschinenbefehls *add x*.

2

Gegeben sind zwei ganze Zahlen *a* und *b* und zwei Methoden in Pseudocode:

```

Methode rechnen1(a, b)
  wenn a gleich b dann
    gib a zurück
  sonst
    gib a + rechnen1(a+1, b) zurück
  endeWenn
endeMethode

Methode rechnen2(a, b)
  setze s gleich 0
  wiederhole solange a kleiner oder gleich b
    erhöhe s um a
    erhöhe a um 1
  endeWiederhole
  gib s zurück
endeMethode

```

b Geben Sie für jede der beiden Methoden an, welche Bedingung die Aufrufparameter *a* und *b* erfüllen müssen, damit die Ausführung der Methode terminiert.

2

Geben Sie begründet an, welche der beiden Methoden rekursiv ist.

(Fortsetzung nächste Seite)

c Ermitteln Sie nachvollziehbar, z. B. durch Angabe der Aufrufsequenz, welches Ergebnis der Aufruf *rechnen1(6, 9)* liefert.

2

Die beiden Methoden *rechnen1* und *rechnen2* liefern dasselbe Ergebnis, sofern sie für die übergebenen Parameter terminieren. Geben Sie die Bedeutung des Rückgabewerts an und beschreiben Sie die unterschiedlichen Vorgehensweisen der beiden Methoden.

d Übertragen Sie die Methode *rechnen2* in ein Programm für eine Registermaschine mit dem oben genannten Befehlssatz. Verwenden Sie dabei die symbolische Adressierung und speichern Sie den Rückgabewert in einer nachvollziehbar benannten Speicherzelle.

4

2 Der Server eines Online-Versandhandels kann mehrere Bestellungen zeitgleich entgegennehmen, wobei jede Bestellung einen eigenen Prozess auslöst.

Das zentrale Objekt *onlineversand* hat Zugriff auf den gesamten Artikelbestand und auf alle Bestellungen. Typische Methoden, die dieses Objekt ausführen kann, sind:

- *anzahlGeben(artikel)* gibt die verfügbare Stückzahl des übergebenen Artikels zurück.
- *bestellen(artikel, anzahl)* fügt den übergebenen Artikel in der übergebenen Anzahl zu der zum Prozess gehörigen Bestellung hinzu und aktualisiert den Artikelbestand.

a Erläutern Sie anhand eines Sequenzdiagramms, welches Problem auftreten kann, wenn mehrere Prozesse gleichzeitig auf das Objekt *onlineversand* zugreifen.

3

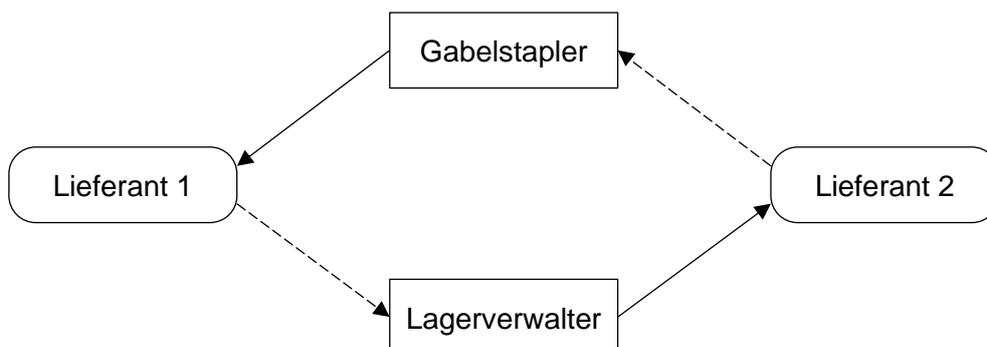
b Nennen Sie ein Konzept, welches das in Teilaufgabe 2a genannte Problem verhindert. Erläutern Sie das Konzept allgemein und im Kontext.

2

3 Im Lager des Online-Versandhandels werden die Artikel von verschiedenen Lieferanten angeliefert. Um die neu eingetroffene Ware korrekt zu erfassen, muss sie mithilfe des vorhandenen Gabelstaplers ins Lager gebracht werden. Außerdem muss die Bestandsliste vom Lagerverwalter aktualisiert werden.

a Betrachten Sie die folgende Situation:

4



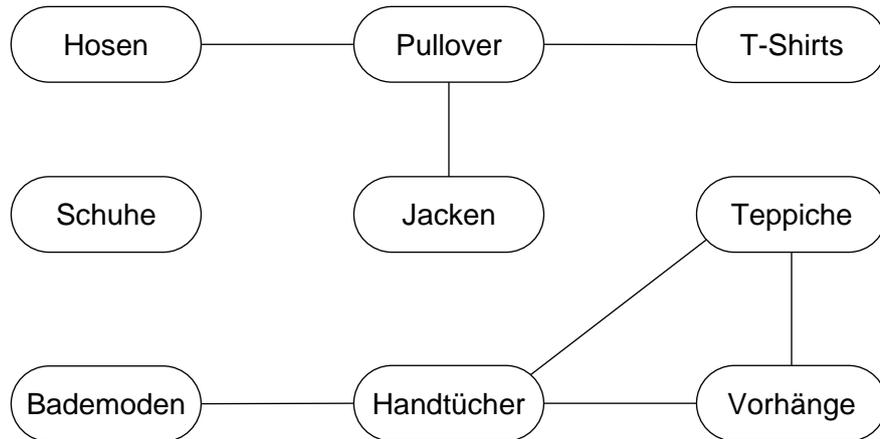
Erläutern Sie den Betriebsmittelzuteilungsgraph im Sachzusammenhang. Beschreiben Sie die vier Coffman-Bedingungen, die erfüllt sein müssen, damit eine Verklemmung entstehen kann, und geben Sie an, woran man im Betriebsmittelzuteilungsgraph eine Verklemmung erkennen kann.

b Geben Sie eine Möglichkeit an, wie die in Teilaufgabe 3a dargestellte Verklemmung aufgelöst werden kann. Stellen Sie den Bezug zu derjenigen Coffman-Bedingung her, die durch Ihre Möglichkeit nicht mehr zutreffen kann.

1

(Fortsetzung nächste Seite)

- 4 Der folgende Graph zeigt einige Warengruppen aus dem Angebot des Online-Versandhandels. Eine Kante repräsentiert, dass Artikel aus diesen Warengruppen häufig miteinander gekauft wurden.



Es soll eine Methode *erreichbareAusgeben(start)* entwickelt werden, die ausgehend von der Warengruppe *start* ausgibt, welche Warengruppen von dieser aus erreichbar sind.

- a Beschreiben Sie im Sachzusammenhang eine mögliche Anwendung dieser Methode. 1
- b Formulieren Sie einen Algorithmus z. B. in Pseudocode für die Methode *erreichbareAusgeben*. 4
- 5 Da die Software des Online-Versandhandels nicht immer wie gewünscht arbeitet und gelegentlich fehlerhafte Bestellungen aufgenommen werden, soll sie anhand von Testverfahren überprüft werden. Zudem soll eine Refaktoriierung des Quelltextes stattfinden.
- a Nennen Sie drei verschiedene Arten von Softwaretests und entscheiden Sie, welche Art von Test für das gegebene Szenario geeignet ist. Begründen Sie Ihre Entscheidung. 3
- b Beschreiben Sie, wozu eine Refaktoriierung des Quelltextes dienen kann. 1
- c Nennen Sie eine weitere Maßnahme, die zur Steigerung der Softwarequalität beiträgt, und erläutern Sie, inwiefern hierdurch eine Qualitätssteigerung erreicht wird. 1

- 1 Die Autowaschanlage „Bitsblank“ bietet Waschprogramme zu ganz bestimmten Preisen an. Die Überprüfung, ob ein Preis (in Cent) zu einem der angebotenen Waschprogramme passt, erfolgt mithilfe einer Klasse AUTOMAT, die ein ganzzahliges Attribut *zustand* und die folgenden Methoden besitzt:

Methode *zustandWechseln*(ZEICHEN eingabe)

```
falls zustand gleich
  0: falls eingabe gleich
    '1': zustand = 2
    '3': zustand = 1
    '5': zustand = 1
    '8': zustand = 3
    sonst: zustand = 6
  endeFalls
  1: wenn eingabe gleich '9' dann
    zustand = 4
  sonst
    zustand = 6
  endeWenn
  2: falls eingabe gleich
    '1': zustand = 3
    '3': zustand = 1
    sonst: zustand = 6
  endeFalls
  3: wenn eingabe gleich '5' dann
    zustand = 4
  sonst
    zustand = 6
  endeWenn
  4: wenn eingabe gleich '0' dann
    zustand = 5
  sonst
    zustand = 6
  endeWenn
  sonst: zustand = 6
endeFalls
endeMethode
```

(Fortsetzung nächste Seite)

```

Methode pruefen(ZEICHENKETTE preis)
zustand = 0
zähle i von 0 bis preis.laenge() - 1
    zustandWechseln(preis.zeichenAn(i))
endeZähle
wenn zustand gleich 5 dann
    gib wahr zurück
sonst
    gib falsch zurück
endeWenn
endeMethode

```

Dabei verwendet die Methode *pruefen* die folgenden Methoden der Klasse ZEICHENKETTE:

- *laenge()* liefert die Länge der Zeichenkette.
- *zeichenAn(n)* liefert das Zeichen an der n-ten Position der Zeichenkette, wobei die Zählung bei 0 beginnt.

Ein Preis p für ein Waschprogramm ist genau dann gültig, wenn *pruefen*(p) *wahr* zurückgibt.

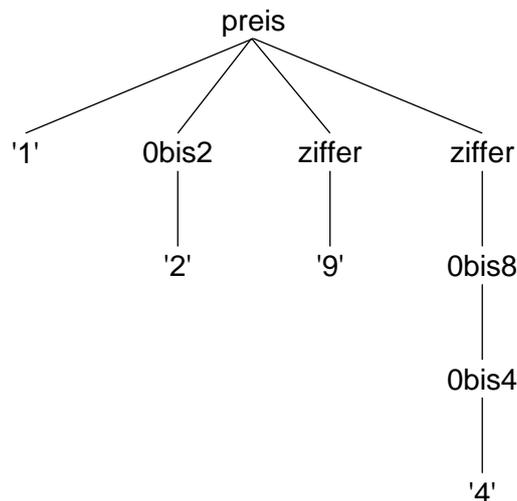
Die Menge aller gültigen Preise bildet die formale Sprache L_1 .

a Zeichnen Sie das Zustandsübergangsdiagramm eines endlichen Automaten, der genau die gültigen Preise akzeptiert. Zur Vereinfachung können Sie dabei auf den Fehlerzustand verzichten. Geben Sie den zum Fehlerzustand gehörenden Attributwert an. 5

b Die Sprache L_1 umfasst lediglich 5 Wörter. Geben Sie diese an. 3

c Um flexibler in der Preisgestaltung zu sein, sollen alle Preise von 350 bis 1390 zulässig sein. Die Menge aller nun gültigen Preise bildet die formale Sprache L_2 . 4

Beispielsweise kann mit nachstehendem Ableitungsbaum begründet werden, dass 1294 ein gültiger Preis ist.



Dabei sind *preis*, *ziffer*, *0bis2*, *0bis4* und *0bis8* geeignete Nichtterminale. Als Terminale dienen ausschließlich die Ziffern 0 bis 9.

Notieren Sie unter Verwendung obiger und ggf. weiterer Nichtterminale die Produktionsregeln der formalen Sprache L_2 in EBNF.

(Fortsetzung nächste Seite)

- d** Um die Preise in einer LED-Laufschrift anzuzeigen, sollen zur optischen Abgrenzung nach dem Preis so viele Zeichen * folgen, wie der Preis Stellen hat, wobei beliebige Preise (in Cent) angezeigt werden sollen. 4
- Alle ganzzahligen Preise größer 0 gefolgt von so vielen Zeichen *, wie der Preis Stellen hat, bilden die formale Sprache L_3 . Zu L_3 gehören z. B. 3^* und 1234567^{*****} .
- Notieren Sie unter Verwendung eines Syntaxdiagramms die Produktionsregeln der formalen Sprache L_3 .
- Entscheiden Sie auch, ob die Sprache L_3 regulär ist, und begründen Sie Ihre Entscheidung.
- e** Die Betreiber der Autowaschanlage „Bitsblank“ haben die Software „Bitsblanksoft“ bei einem Softwareunternehmen in Auftrag gegeben. Diese Software enthält u. a. Komponenten zur Kundenverwaltung, zur Steuerung der Waschanlage und zur Abrechnung. Die Komponente zur Steuerung ist nun fertiggestellt. 2
- Beschreiben Sie, zu welchem Zeitpunkt die Komponente zur Steuerung beim Betreiber der Waschanlage ausgeliefert wird, wenn das Softwareunternehmen bei der Entwicklung
- i gemäß dem Wasserfallmodell vorgeht,
 - ii eine agile Vorgehensweise verwendet.
- 2** Ein einfacher Algorithmus zur Bestimmung einer Liste aller Primzahlen von 2 bis n ist „das Sieb des Eratosthenes“. Der Algorithmus kann wie folgt definiert werden:
- ```

Methode primzahlenErmitteln(n)
 schreibe alle Zahlen von 2 bis n in eine Liste
 zähle i von 2 bis n
 zähle k von 2 bis n
 wenn $i * k$ kleiner oder gleich n dann
 streiche die Zahl $i * k$, falls vorhanden, aus der Liste
 endeWenn
 endeZähle
 endeZähle
endeMethode

```
- a** Beschreiben Sie die Funktionsweise des Algorithmus am Beispiel  $n = 9$ . 2
- b** Stellen Sie eine Möglichkeit dar, wie der gegebene Algorithmus so verbessert werden kann, dass sich die Laufzeit verringert. 2
- c** Auf einem Rechner terminiert der Algorithmus für  $n = 10^5$  nach 20 Sekunden. Berechnen Sie, wie lange die Ausführung für  $n = 10^8$  dauern würde unter der Annahme 4
- i eines linearen Laufzeitverhaltens,
  - ii eines quadratischen Laufzeitverhaltens.
- Geben Sie begründet das Laufzeitverhalten des gegebenen Algorithmus an.

(Fortsetzung nächste Seite)

3 Vereinfachend wird eine Methode als Zeichenkette *quelltext* angesehen. Die Parameter, die der Methode übergeben werden müssen, werden ebenfalls als Zeichenkette *daten* interpretiert.

Mit *quelltext(daten)* wird dann der Aufruf der von *quelltext* repräsentierten Methode mit den Parametern von *daten* verstanden. Die Ausführung der Methode kann terminieren oder nicht terminieren.

Betrachtet wird nun die folgende Methode:

```
Methode isthaltend(quelltext, daten)
 wenn quelltext(daten) terminiert dann
 gib wahr zurück
 sonst
 gib falsch zurück
 endeWenn
endeMethode
```

Beim Halteproblem wird gefragt, ob diese Methode für alle möglichen Eingaben feststellen kann, ob *quelltext(daten)* terminiert. Um zu beweisen, dass die Methode dies nicht leisten kann, entwirft man nachstehende Testmethode *test*, der eine Zeichenkette *eingabe* übergeben wird.

```
Methode test(eingabe)
 wiederhole solange isthaltend(eingabe, eingabe)
 endeWiederhole
endeMethode
```

Auch die Methode *test* kann als Zeichenkette aufgefasst werden.

Zeigen Sie, dass sich beim Aufruf von *isthaltend(test, test)* stets ein Widerspruch ergibt.

1 Die Betreiberin eines Online-Shops möchte auf Basis der Parameter Zuckeranteil, Fettanteil, Ballaststoffanteil und Eiweißanteil Lebensmittel mithilfe eines künstlichen neuronalen Netzes als gesund oder ungesund einstufen. Das künstliche neuronale Netz soll mit Backpropagation trainiert werden. Dazu liegt eine Menge an Datensätzen mit konkreten Werten vor, für die Fachleute zuvor eine Einstufung vorgenommen haben.

a Geben Sie an, welches Verfahren maschinellen Lernens in der beschriebenen Situation zum Einsatz kommt, und begründen Sie, dass sich dieses prinzipiell zur Lösung des gegebenen Problems eignet.

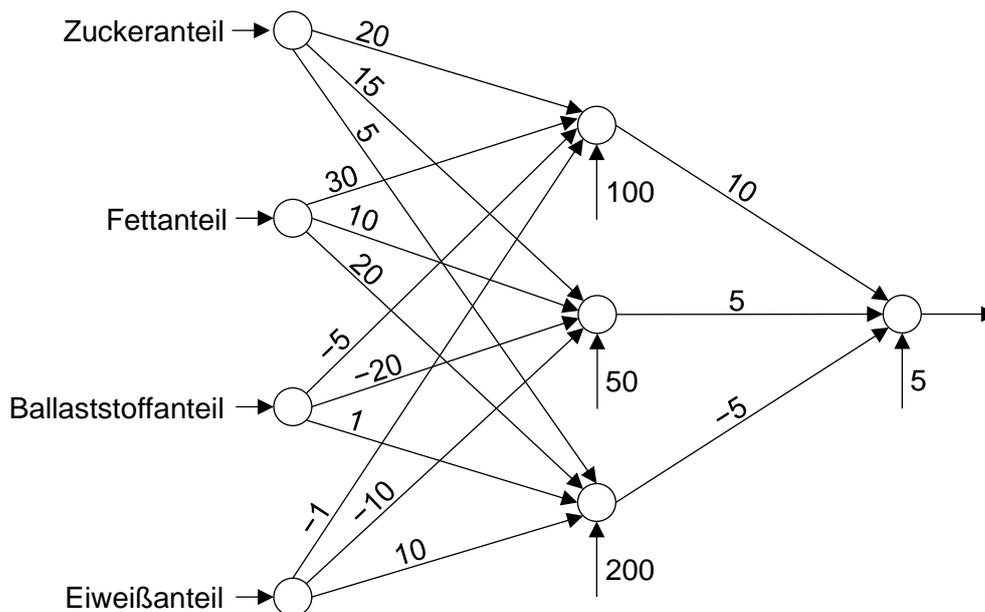
1

b Beschreiben Sie allgemein, welche Rolle Forward Propagation und Backpropagation beim Lernvorgang und beim späteren produktiven Einsatz des künstlichen neuronalen Netzes spielen.

2

c Es kommt das folgende künstliche neuronale Netz zum Einsatz, wobei das Neuron der Ausgangsbeschriftung feuern soll, wenn das Lebensmittel als ungesund gelabelt ist. Als Aktivierungsfunktion wird in der Eingabeschicht die Identitätsfunktion und ansonsten die Heaviside-Funktion verwendet. Das Netz befindet sich anfangs in einem untrainierten Zustand, in dem die Gewichte und die Schwellenwerte zufällig gesetzt sind.

3



Berechnen Sie durch Forward Propagation die Aktivierungen am Ausgang des künstlichen neuronalen Netzes für folgende Lebensmittel und geben Sie an, ob sie als gesund oder ungesund klassifiziert werden. Die Werte sind in Prozent angegeben.

- Lebensmittel A (20, 30, 10, 5)
- Lebensmittel B (5, 5, 50, 20)

d Das trainierte künstliche neuronale Netz zur Klassifizierung von Lebensmitteln liefert auf den Trainingsdaten eine Vorhersagequote von 100 %. Mit Testdaten wird jedoch eine wesentlich geringere Vorhersagequote erzielt. Erklären Sie, wie ein solches Phänomen zustande kommen kann.

2

(Fortsetzung nächste Seite)

- 2 Ein Lebensmittelproduzent führt eine kurze Umfrage bei seiner Kundschaft durch, bei der diese jeweils auf einer Skala von 0 bis 100 angeben soll, wie wichtig für die Kaufentscheidung ausgewogene Nährwerte des Produktes bzw. die Verwendung von Rohstoffen aus der Region sind.
- a Die Ergebnisse der Umfrage werden in einer Web-Datenbank gespeichert. Um den Mitarbeitenden Zugriff auf die Daten zu ermöglichen, werden für sie automatisiert Zugangsdaten angelegt. Diese bestehen aus der innerhalb des Unternehmens bekannten Mitarbeiterkennung als Nutzernamen und 12345 als initialem Passwort. Bei der Erstanmeldung werden die Mitarbeitenden dann aufgefordert, ihr Passwort zu ändern. Nennen Sie zwei hier relevante Gefährdungskategorien und erläutern Sie diese im Sachzusammenhang. 2
- b Nennen Sie zwei Schutzziele der Informationssicherheit, die Sie im Kontext der Umfrage für besonders wichtig halten, und begründen Sie Ihre Entscheidung. 2
- c Erläutern Sie den Grundansatz des k-Means-Algorithmus und beschreiben Sie im Sachzusammenhang, zu welchem Zweck dieser Algorithmus vom Lebensmittelproduzenten in der Analyse der Kundenantworten eingesetzt werden könnte. 2
- d Gehen Sie davon aus, dass der k-Means-Algorithmus zunächst mit zwei Clusterzentren  $C_1$  und  $C_2$  durchgeführt werden soll. Diese wurden zu Beginn wie folgt zufällig positioniert:  $C_1(30|40)$  und  $C_2(70|70)$ .  
Geben Sie begründet an, welchem Clusterzentrum ein Datenpunkt D mit den Werten (60|50) zugeordnet werden muss. Verwenden Sie zur Berechnung der Distanz zweier Punkte den geometrischen Abstand. 3
- e Im Folgenden soll der k-Means-Algorithmus für das gegebene Problem der Kaufentscheidungen umgesetzt werden. 7
- Die Daten aus der Umfrage sowie die Clusterzentren werden durch Objekte der Klasse PUNKT mit den ganzzahligen Attributen *nährwert* und *regionalität* repräsentiert. Die Klasse PUNKT verfügt über eine Methode *distanzGeben(PUNKT p)*, die den geometrischen Abstand vom ausführenden Punkt-Objekt zum Punkt *p* liefert. Sie können davon ausgehen, dass die Klasse PUNKT einschließlich der Methoden zum Geben und Setzen der Attributwerte bereits vollständig implementiert ist.
- Außerdem sind folgende indizierte Datenstrukturen gegeben: Das Feld `PUNKT[] daten` referenziert bereits die *n* Datenpunkte aus der Umfrage und das Feld `PUNKT[] zentren` die *k* Clusterzentren. Im Feld `GANZZAHL[] zuordnung` wird gespeichert, welcher Datenpunkt welchem Clusterzentrum zugeordnet wird. Dabei bedeutet beispielsweise `zuordnung[7] = 3`, dass der mit dem Wert 7 indizierte Datenpunkt dem Clusterzentrum mit Index 3 zugeordnet wird.
- Implementieren Sie folgende Methoden:
- *klassifizieren()* ordnet jedem Datenpunkt ein Clusterzentrum zu.
  - *zentrenAktualisieren()* aktualisiert für jedes Clusterzentrum anhand der ihm zugeordneten Datenpunkte die Werte seiner Attribute.

(Fortsetzung nächste Seite)

3 In einem App-Store findet sich die folgende Beschreibung einer App „Dr. KI“:

**„Dr. KI“ als Ernährungsberater**

Die Smartphone-App „Dr. KI“ misst mithilfe von Wearables Körperdaten (Schlafrythmus, Bewegungsverhalten, Puls etc.) und gibt individuelle Ratschläge zur Ernährung und Lebensweise. Die App basiert auf einem künstlichen neuronalen Netz. Sie wurde anhand einer Gruppe von 20 000 Stammgästen von Fitnessstudios trainiert und auf Zuverlässigkeit überprüft.

- a Es kommt häufig vor, dass „Dr. KI“ Ernährungstipps gibt, die nicht zu den Nutzerinnen und Nutzern der App passen. Nehmen Sie aus informatischer Sicht Stellung, wie es zu diesem Problem kommen kann. 1
- b Die Entwickler der App versichern, das in Teilaufgabe a beschriebene Problem durch Anpassung der Hyperparameterwerte zu lösen. Geben Sie zwei Hyperparameter an und beurteilen Sie, ob sich das Problem lösen lässt, indem deren Werte geändert werden. 2
- c Diskutieren Sie im Kontext des Beispiels „Dr. KI“ anhand von je zwei Aspekten Chancen und Risiken eines solchen KI-Systems. 3

30