



# **Informatik**

## **Leistungskurs**

### **Beispielset 1**

**Auswahlverfahren:**        **siehe Seite 2**

**Einlese- und Auswahlzeit:**    **30 Minuten**

**Bearbeitungszeit:**            **240 Minuten**

<b>Erlaubte Hilfsmittel:</b>	<b>Wörterbuch zur deutschen Rechtschreibung Hessisches Datenschutzgesetz</b>
<b>Sonstige Hinweise:</b>	<b>keine</b>

## **Auswahlverfahren**

Die Prüfungsaufgaben sind in die zwei Gruppen A und B eingeteilt.

- Die beiden Aufgaben der Gruppe A sind Pflichtaufgaben.
- Die Prüfungsteilnehmerinnen und Prüfungsteilnehmer entscheiden sich für eine der beiden Wahlaufgaben aus Gruppe B.

Insgesamt sind drei Aufgaben zu bearbeiten, zwei aus Gruppe A und eine aus Gruppe B.

## I. Thema und Aufgabenstellung

### Gruppe A Pflichtaufgaben

### Gruppe B Auswahl einer Aufgabe durch die Prüfungsteilnehmerinnen und Prüfungsteilnehmer

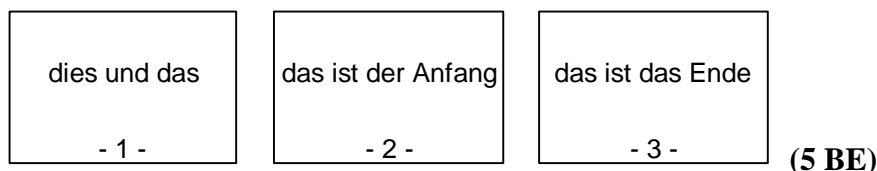
## Aufgaben

### Gruppe A

#### A 1 Stichwortverzeichnis (Delphi, Pascal oder Java, $\Sigma$ 35 BE)

Germanisten möchten zur Durchführung von Sprach- und Stil-Untersuchungen zu einem Buch ein Wortverzeichnis erstellt haben. Darin sollen alle im Buch vorkommende Wörter alphabetisch geordnet enthalten sein und bei jedem Wort sollen in aufsteigender Reihenfolge die Seitenzahlen stehen, bei denen das Wort vorkommt. Um trotz der hohen Zahl von Wörtern das Wortverzeichnis schnell erstellen und verwalten zu können, soll es als binärer Suchbaum aufgebaut werden.

a) Zeichnen Sie den binären Suchbaum für den folgenden auf drei Seiten verteilten Text.



- b) Modellieren Sie Attribute und Methoden für einen binären Suchbaum. Die Knoten sollen jeweils ein Wort enthalten und eine geeignete Datenstruktur mit den zugehörigen Seitenzahlen. Stellen Sie ihr Modell als UML-Klassendiagramm dar. **(6 BE)**
- c) Ein Wort und eine zugehörige Seitenzahl sollen bereit stehen. Implementieren Sie Methoden zum Einfügen dieser Daten in den binären Suchbaum, samt erforderlicher Konstruktor. **(12 BE)**
- d) Nach der automatischen Erstellung des Wortverzeichnisses sollen unwichtige Wörter wie der, die, das gelöscht werden können. Beschreiben Sie ein Verfahren, mit dem man einen bestimmten Knoten eines binären Suchbaums möglichst effizient entfernen kann. Begründen Sie das Verfahren. **(6 BE)**
- e) Implementieren Sie einen Algorithmus zur Ermittlung eines Wortes, das am häufigsten vorkommt. **(6 BE)**

## A 2 Datenbank Kino (Σ 35 BE)

Ein Kino verfügt über mehrere Säle mit unterschiedlicher Anzahl von Sitzplätzen. Die Sitzplätze in einem Saal sind zu Verwaltungszwecken aufsteigend nummeriert. Für jeden Sitzplatz sind auch die Kategorie (Loge bzw. Parkett), die Reihe und seine spezielle Platznummer in der Reihe angegeben. In den Sälen finden die Vorstellungen statt, die durch eindeutige Vorstellungsnummern identifiziert werden. Zu jeder Vorstellung soll die Gesamtzahl der bisher verkauften Kinokarten gespeichert werden. Das Kinoprogramm gibt alle Vorstellungen nach Datum, Uhrzeit und gezeigtem Film an. Neben den Filmtiteln sind Regisseur, Dauer, Kategorie (Action, Komödie, ...) und FSK-Altersangabe wichtig. Auf jeder Kinokarte werden Reihe und Platznummer des gebuchten Sitzplatzes angegeben.

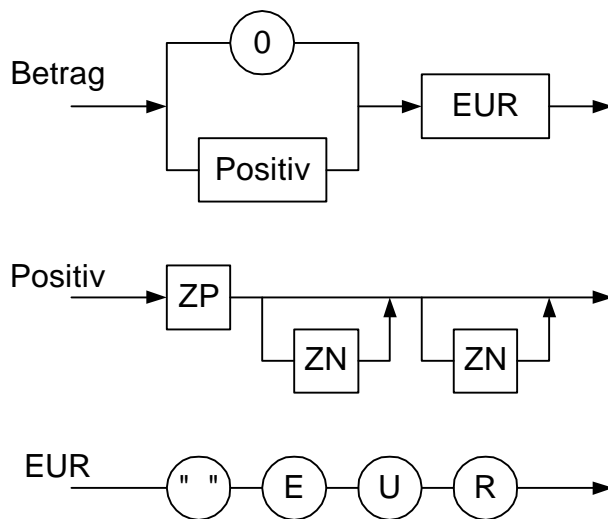
- a) Modellieren Sie eine Datenbank für das Kino als ER-Diagramm, analysieren und beschreiben Sie die im ER-Diagramm vorkommenden Beziehungen. **(10 BE)**
- b) Übertragen Sie das ER-Diagramm in ein optimiertes Relationenmodell und erläutern Sie das Primärschlüsselproblem der Sitzplatztabelle. **(4 BE)**
- c) Erläutern Sie sowohl SQL-technisch als auch anwendungsbezogen die beiden folgenden SQL-Anweisungen: **(6 BE)**
- c1) `SELECT Datum, Uhrzeit, Titel, FSK  
FROM Vorstellung, Film  
WHERE Vorstellung.FilmID = Film.FilmID AND Datum = #10/04/2004#  
ORDER BY Uhrzeit`
- c2) `SELECT VorstellungID, Datum, Titel  
FROM Vorstellung, Film  
WHERE Vorstellung.FilmID = Film.FilmID AND  
Verkauft = (  
    SELECT MAX(Verkauft)  
    FROM Vorstellung, Film  
    WHERE Vorstellung.FilmID = Film.FilmID AND Kategorie = 'Action' )`
- d) Formulieren Sie SQL-Anweisungen für folgende Fragestellungen: **(9 BE)**
- d1) Gesucht ist eine Liste aller Abenteuer-Filme die im Monat Mai laufen.
- d2) Gesucht ist eine Liste der nicht ausverkauften Vorstellungen von *Herr der Ringe III*.
- d3) Gesucht ist eine Liste mit den Gesamtzahlen aller verkauften Kinokarten pro Film für die Filme mit mindestens 1000 Besuchern.
- e) Über das Internet können Einzelplatz-Reservierungen von Vorstellungen vorgenommen werden. Man gibt dabei seinen Namen und die Sitzplatz-Kategorie an und erhält als Bestätigung eine Reservierungsnummer, die man an der Kinokasse vorlegen muss. Beschreiben Sie die benötigte Datenbank-Erweiterung und bewerten Sie das Online-Reservierungssystem unter technischen und gesellschaftlichen Aspekten. **(6 BE)**

## Gruppe B

### B 1 Geldbeträge (Σ 30 BE)

Im Folgenden geht es um eine Grammatik für Geldbeträge in der Währung Euro. Dabei wird die im internationalen Zahlungsverkehr verwendete Abkürzung EUR benutzt.

- a) Erläutern Sie die drei Syntaxdiagramme für Geldbeträge, beschreiben Sie die durch sie definierte Sprache und erweitern Sie die Syntaxdiagramme so, dass optional zweistellige Kommabeträge möglich sind. (6 BE)



ZP steht für die Ziffern von 1 bis 9 und ZN für die Ziffern von 0 bis 9.

- b) Zeichnen Sie das Zustandsdiagramm eines endlichen Automaten, der Geldbeträge nach Aufgabe a mit optionalem zweistelligen Kommabetrag erkennt. (6 BE)
- c) Übersetzen Sie das Zustandsdiagramm in eine äquivalente Grammatik und geben Sie eine Ableitung für das Wort 546,80 EUR an. (7 BE)
- d) Erläutern Sie den folgenden Akzeptor: (4 BE)

```

akzeptiere_positiv(Ein, Aus):-
    akzeptiere_zp(Ein, L1),
    akzeptiere_zn(L1, L2),
    akzeptiere_zn(L2, Aus).

akzeptiere_zp([K|R], R):-
    member(K, ['1', '2', '3', '4', '5', '6', '7', '8', '9']).

akzeptiere_zn([K|R], R):-
    member(K, ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']).
akzeptiere_zn(R, R).
  
```

- e) Erweitern Sie Aufgabe d zu einem Akzeptor für Betrag mit optionalem zweistelligen Kommabetrag. (7 BE)

**B 2 Kellerautomaten ( $\Sigma$  30 BE)**

Eine formale Sprache L sei durch die folgende Grammatik gegeben:

Nicht-Terminal = {S, B, C}, Terminal = {a, b, c,  $\varepsilon$ }, Startzeichen S,  $\varepsilon$  = leeres Wort

Produktionen:

$S \rightarrow aSa$	$S \rightarrow bBb$	$S \rightarrow cC$
$B \rightarrow bBb$	$B \rightarrow cC$	
$C \rightarrow cC$	$C \rightarrow \varepsilon$	

- a) Bilden Sie einige Worte der Sprache L und beschreiben Sie diese Sprache. Begründen Sie, dass diese Sprache nicht durch einen endlichen Automaten erkannt werden kann und ordnen Sie sie in die Chomsky-Hierarchie ein. **(7 BE)**
- b) Stellen Sie einen Kellerautomaten K dar, der alle Worte der Sprache L erkennt. **(15 BE)**
- c) Der Kellerautomat K soll in einer Programmiersprache (z. B. Delphi, Pascal oder Java) simuliert werden. Beschreiben Sie geeignete Datenstrukturen und Zugriffsoperationen auf diese Datenstrukturen. **(5 BE)**
- d) Die Sprache  $T = \{a^n b^n c b^n a^n\}$  mit  $n \in \{1, 2, 3, \dots\}$  ist eine Teilmenge von L. Erläutern Sie, warum die Sprache T nicht mit einem Kellerautomaten überprüft werden kann. **(3 BE)**

## Korrektur- und Bewertungshinweise - nicht für den Prüfungsteilnehmer bestimmt -

### II. Erläuterungen, Lösungshinweise, Bewertung und Beurteilung

#### A 1 Stichwortverzeichnis – Delphi/Pascal

##### Aufgabenart:

Die Aufgabe gehört zum Halbjahresthema 12.1 „Objektorientierte Modellierung“. Schwerpunkt ist die Arbeit mit den Datenstrukturen binärer Suchbaum und lineare Liste. Bekannte Algorithmen sollen wiedergegeben und neue Algorithmen entwickelt werden. Ein Teil der Algorithmen ist in einer Programmiersprache zu implementieren.

##### Voraussetzungen des Lehrplans:

Zur Lösung der Aufgabe werden folgende verbindlichen Inhalte des Lehrplans benötigt: Objektmodell, Klassen, lineare Listen, binärer Suchbaum, Repräsentierung und Standardoperationen, rekursive und iterative Verfahren.

##### Bewertungseinheiten

	erwartete Schülerleistung	I	II	III
a	<p>Mehrfach vorkommende Seitenzahlen werden aufgeführt.</p> <pre> graph TD     A((dies 1)) --&gt; B((das 1,2,3,3))     A --&gt; C((und 1))     B --&gt; D((Anfang 2))     B --&gt; E((der 2))     C --&gt; F((ist 2,3))     F --&gt; G((Ende 3)) </pre>	5		
b	<p>Die Knoten enthalten das Wort und eine geordnete lineare Liste der Seitenzahlen. Es ist zweckmäßig drei Klassen zu entwerfen: TBinBaum, TBinKnoten und TListe. Im UML-Diagramm sind auch Attribute und Methoden eingetragen, die sich aus den Teilen c, d und e ergeben. Dies ist an dieser Stelle noch nicht erforderlich, aber zur besseren Darstellung zweckmäßig. Statt der gerichteten Assoziation zwischen TBinKnoten und TListe kann auch eine Aggregation benutzt werden.</p>	3	3	

erwartete Schülerleistung		I	II	III
<pre> classDiagram     class TListe {         Zahl: integer         Next: TListe         c create(...)         einfuegen(...)         gibAnzahl     }     class TBinKnoten {         Wort: string         Seitenzahlen: TListe         links: TBinKnoten         rechts: TBinKnoten         c create(...)     }     class TBinBaum {         Wurzel: TBinKnoten         MaxWort: string         MaxZahl: integer         c create(...)         einfuegen(...)         loeschen(...)         haeufigstesWort(...)     }     TListe --&gt; TBinKnoten     TBinKnoten *-- TBinBaum         </pre>				
c	<p>Mögliche Lösung:</p> <pre> <b>constructor</b> TListe.create(NeueZahl: integer; wohin: TListe); <b>begin</b>     Zahl:= NeueZahl;     next:= wohin; <b>end;</b>  <b>procedure</b> TListe.einfuegen(NeueZahl: integer);     <b>var</b> NeuesElement: TListe; <b>begin</b>     <b>if</b> (next = <b>nil</b>) <b>or</b> (NeueZahl &lt; next.Zahl) <b>then</b>         <b>begin</b>             NeuesElement:= TListe.create(NeueZahl, next);             next:= NeuesElement;         <b>end</b>     <b>else</b> next.einfuegen(NeueZahl); <b>end;</b>  <b>constructor</b> TBinKnoten.create(NeuesWort: string;                                 NeueZahl: integer); <b>begin</b>     wort:= NeuesWort;     SeitenZahlen:= TListe.create(NeueZahl, <b>nil</b>);     links:= <b>nil</b>;     rechts:= <b>nil</b>; <b>end;</b>  <b>procedure</b> TBinBaum.einfuegen(<b>var</b> Wurzel:TBinKnoten;                                 NeuesWort:string; NeueZahl:integer); <b>begin</b>     <b>if</b> wurzel = <b>nil</b> <b>then</b>         wurzel:= TBinKnoten.create(NeuesWort, NeueZahl)     <b>else</b>         <b>if</b> NeuesWort=Wurzel.Wort <b>then</b>             Wurzel.Seitenzahlen.einfuegen(NeueZahl)         <b>else</b>             <b>if</b> NeuesWort&lt;Wurzel.Wort <b>then</b>                 einfuegen(Wurzel.links,NeuesWort,NeueZahl)             <b>else</b>                 einfuegen(Wurzel.rechts,NeuesWort,NeueZahl);         <b>end</b>     <b>end;</b>         </pre>		12	
d	<p>Beschreibung der Löschsens unter Berücksichtigung der drei Fälle:</p> <p>1. Löschen eines Blatts</p> <p>Ein Blatt kann gelöscht werden, indem man im Vater des Blatts den Zeiger auf das Blatt auf <b>null</b> setzt. Existiert kein Vater, so ist das Blatt die Wurzel und man setzt die Wurzel auf <b>null</b>.</p>	1		



	erwartete Schülerleistung	I	II	III
	<p>2. Löschen eines Knotens mit einem Teilbaum Man ersetzt den Inhalt des zu löschenden Knotens durch die Wurzel des einen Teilbaums, der am Knoten dran hängt.</p> <p>3. Löschen eines Knotens mit zwei Teilbäumen Die beste Methode besteht darin, den Inhalt des Knotens durch den Inhalt eines geeigneten anderen Knotens zu ersetzen. Geeignet ist der am weitesten rechts gelegene Knoten des linken Teilbaums, da sein Inhalt größer ist als der aller Knoten des linken Teilbaums und kleiner als der aller Knoten des rechten Teilbaums. Dieser Knoten hat also genau die gleichen Ordnungseigenschaften wie der zu löschende Knoten. Außerdem kann er nach Fall 2 leicht gelöscht werden, da er per Definition keinen rechten Teilbaum hat. Entsprechende Überlegungen gelten für den am weitesten links gelegenen Knoten des rechten Teilbaums.</p>	1	4	
e	<p>Das gesuchte Wort MaxWort und die Häufigkeit MaxZahl, mit der es vorkommt, werden als Attribute von TBinBaum angesehen und beim Erzeugen des Objekts mit den Werten '' und 0 versehen. Der Baum wird mit dem LWR-Verfahren durchlaufen. In jedem Knoten wird MaxZahl mit der Anzahl der Seitenzahlen verglichen und gegebenenfalls durch diese ersetzt. In diesem Falle wird auch MaxWort ersetzt. Bei gleicher Anzahl wird MaxWort nicht ersetzt.</p> <pre> function TListe.gibAnzahl: integer; begin     if next = nil     then Result := 1     else Result := next.gibAnzahl + 1; end;  constructor TBinBaum.create; begin     Wurzel := TBinKnoten.create('', 0);     MaxWort := '';     MaxAnzahl := 0; end;  procedure TBinBaum.haeufigstesWort(Wurzel:TBinKnoten); var Anzahl: integer; begin     if Wurzel &lt;&gt; nil then     begin         haeufigstesWort(Wurzel.links);         Anzahl := Wurzel.Seitenzahlen.gibAnzahl;         if Anzahl &gt; MaxAnzahl then         begin             MaxAnzahl := Anzahl;             MaxWort := Wurzel.Wort;         end;         haeufigstesWort(Wurzel.rechts);     end; end; </pre>			6
	Σ 35 BE	10	19	6

## A 1 Stichwortverzeichnis – Java

### Aufgabenart:

Die Aufgabe gehört zum Halbjahresthema 12.1 „Objektorientierte Modellierung“. Schwerpunkt ist die Arbeit mit den Datenstrukturen binärer Suchbaum und lineare Liste. Bekannte Algorithmen sollen wiedergegeben und neue Algorithmen entwickelt werden. Ein Teil der Algorithmen ist in einer Programmiersprache zu implementieren.

### Voraussetzungen des Lehrplans:

Zur Lösung der Aufgabe werden folgende verbindlichen Inhalte des Lehrplans benötigt: Objektmodell, Klassen, lineare Listen, binärer Suchbaum, Repräsentierung und Standardoperationen, rekursive und iterative Verfahren.

### Bewertungseinheiten

	erwartete Schülerleistung	I	II	III
a	<p>Darstellung des Wortverzeichnis für den gegebenen Text als binärer Suchbaum</p> <pre> graph TD     A((dies 1)) --&gt; B((das 1,2,3,3))     A --&gt; C((und 1))     B --&gt; D((Anfang 2))     B --&gt; E((der 2))     C --&gt; F((ist 2,3))     F --&gt; G((Ende 3))         </pre>	5		
b	<p>Modellierung einer geeigneten Datenstruktur und Darstellung als Klassendiagramm Die Knoten enthalten das Wort und einen Vector mit den Seitenzahlen. Im UML-Diagramm sind auch Attribute und Methoden eingetragen, die sich aus den Teilen c, d und e ergeben. Dies ist an dieser Stelle noch nicht erforderlich, aber zur besseren Darstellung zweckmäßig. Statt der gerichteten Assoziation kann zwischen BinKnoten und Vector auch eine Aggregation modelliert werden.</p>	3	3	
c	<p>Implementierung eines Konstruktors BinKnoten und der einfügen-Methoden</p> <pre> public BinKnoten(String wort, int seitenzahl) {     this.wort = wort;     seitenzahlen = new Vector(); }         </pre>		12	

erwartete Schülerleistung		I	II	III
	<pre> seitenzahlen.add(new Integer(seitenzahl)); links = null; rechts = null; }  private void einfuegenBaum(BinKnoten meinBaum, String wort,                            int seitenzahl) {     int vergleich = wort.compareTo(meinBaum.wort);     if (vergleich &lt; 0)         if (meinBaum.links == null)             meinBaum.links = new BinKnoten(wort, seitenzahl);         else             einfuegenBaum(meinBaum.links, wort, seitenzahl);     else if (vergleich &gt; 0)         if (meinBaum.rechts == null)             meinBaum.rechts = new BinKnoten(wort, seitenzahl);         else             einfuegenBaum(meinBaum.rechts, wort, seitenzahl);     else // vergleich == 0         meinBaum.seitenzahlen.add(new Integer(seitenzahl)); }  public void einfuegen(String wort, int seitenzahl) {     if (wurzel == null)         wurzel = new BinKnoten(wort, seitenzahl);     else         einfuegenBaum(wurzel, wort, seitenzahl); } </pre>			
d	<p>Beschreibung der Löschens unter Berücksichtigung der drei Fälle:</p> <p>1. Löschen eines Blatts Ein Blatt kann gelöscht werden, indem man im Vater des Blatts den Zeiger auf das Blatt auf <b>null</b> setzt. Existiert kein Vater, so ist das Blatt die Wurzel und man setzt die Wurzel auf <b>null</b>.</p> <p>2. Löschen eines Knotens mit einem Teilbaum Man ersetzt den Inhalt des zu löschenden Knotens durch die Wurzel des einen Teilbaums, der am Knoten dran hängt.</p> <p>3. Löschen eines Knotens mit zwei Teilbäumen Die beste Methode besteht darin, den Inhalt des Knotens durch den Inhalt eines geeigneten anderen Knotens zu ersetzen. Geeignet ist der am weitesten rechts gelegene Knoten des linken Teilbaums, da sein Inhalt größer ist als der aller Knoten des linken Teilbaums und kleiner als der aller Knoten des rechten Teilbaums. Dieser Knoten hat also genau die gleichen Ordnungseigenschaften wie der zu löschende Knoten. Außerdem kann er nach Fall 2 leicht gelöscht werden, da er per Definition keinen rechten Teilbaum hat. Entsprechende Überlegungen gelten für den am weitesten links gelegenen Knoten des rechten Teilbaums.</p>	1	1	4
e	<p>Beschreibung und Implementierung des gesuchten Algorithmus</p> <pre> private BinKnoten am_haeufigstenLWR(BinKnoten einKnoten) {     if (einKnoten == null)         return null;     else {         BinKnoten links = am_haeufigstenLWR(einKnoten.links);         BinKnoten rechts = am_haeufigstenLWR(einKnoten.rechts);         if ((links != null) &amp;&amp;             (links.seitenzahlen.size() &gt; </pre>			6

	erwartete Schülerleistung	I	II	III
	<pre>         einKnoten.seitenzahlen.size()))         einKnoten = links;         if ((rechts != null) &amp;&amp;             (rechts.seitenzahlen.size() &gt;              einKnoten.seitenzahlen.size()))             einKnoten = rechts;         return einKnoten;     } }  public BinKnoten am_haeufigsten() {     return am_haeufigstenLWR(wurzel); } </pre>			
	<b>Σ 35 BE</b>	10	19	6

## A 2 Datenbank Kino

### Aufgabenart:

In der Aufgabe sind die Modellierung einer Datenbank als ER-Diagramm, die Umsetzung in das Relationenmodell, die Analyse und Konstruktion von SQL-Anweisungen sowie die Analyse von Web-Datenbanken gefordert.

### Voraussetzungen des Lehrplans

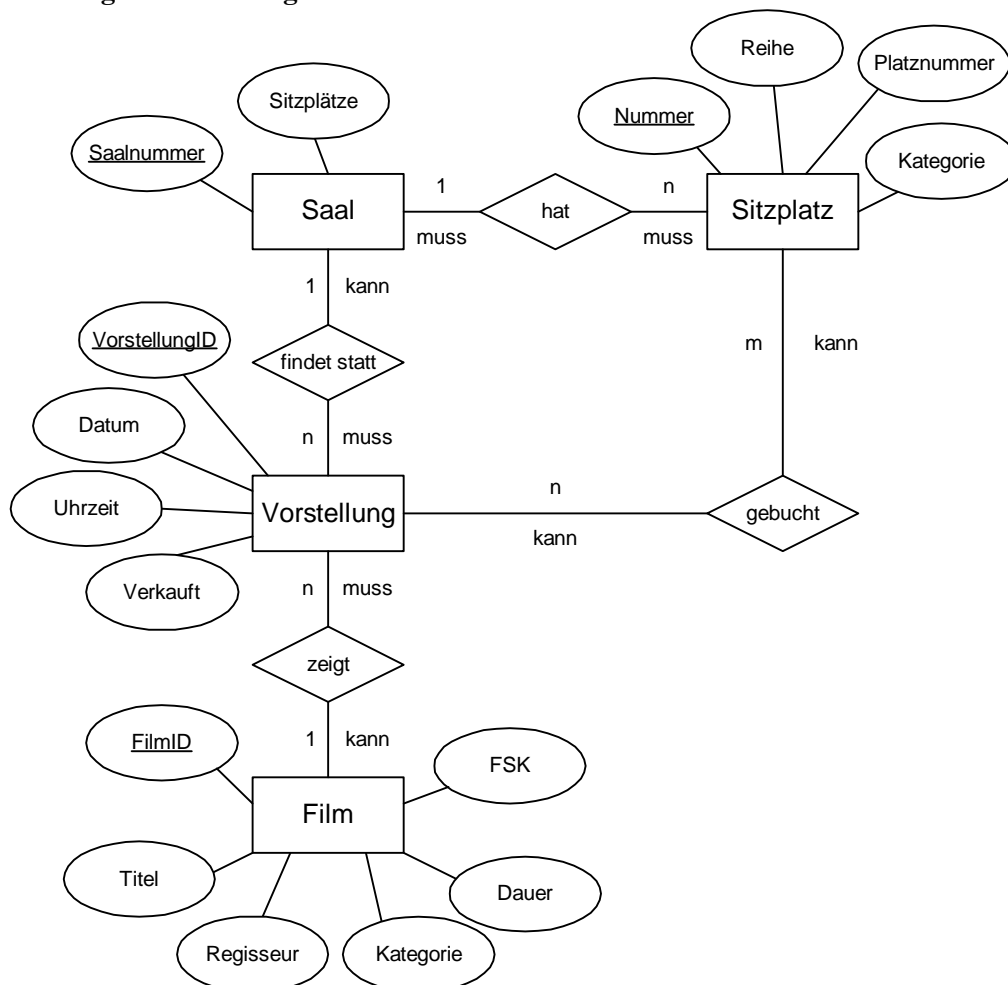
Von den verbindlichen Inhalten des Lehrplans werden zur Lösung der Aufgabe benötigt: die Modellierung von Mini-Welten mit ER-Diagrammen, die Umsetzung eines ER-Diagramms in das Relationenmodell, die Abfragesprache SQL, Datensicherheit und Datenschutz.

### Bewertungseinheiten

	erwartete Schülerleistung	I	II	III
a	<p>Das zu entwerfende ER-Diagramm muss die vier Objekttypen <i>Saal</i>, <i>Sitzplatz</i>, <i>Vorstellung</i> und <i>Film</i> enthalten, sowie die vier Beziehungen <i>findet statt</i>, <i>hat</i>, <i>gebucht</i> und <i>zeigt</i>. Primärschlüssel sind zu unterstreichen, bei der Gestaltung der Attribute gibt es Freiheiten. Die im Text genannten Attribute müssen vorhanden sein.</p> <p>ER-Diagramm siehe unten</p> <p>Kardinalität und Optionalität sind im ER-Diagramm dargestellt. Diese sind sprachlich zu erläutern, z. B. ein Film kann in mehreren Vorstellungen gezeigt werden, daher Kardinalität n, muss aber nicht gezeigt werden, daher Kann-Beziehung. Eine Vorstellung muss genau einen Film zeigen, daher Kardinalität 1 und Muss-Beziehung.</p>	3	3	
b	<p>Saal(<u>Saalnummer</u>, Sitzplätze)</p> <p>Sitzplatz(<u>Nummer</u>, Reihe, Platznummer, Kategorie, <math>\uparrow</math>Saalnummer)</p> <p>Film(<u>FilmNr</u>, Titel, Regisseur, Kategorie, Dauer, FSK)</p> <p>Vorstellung(<u>VorstellungID</u>, Datum, Uhrzeit, Verkauft, <math>\uparrow</math>Saalnummer, <math>\uparrow</math>FilmID)</p> <p>Gebucht(<math>\uparrow</math>Nummer, <math>\uparrow</math>VorstellungID)</p> <p>Primärschlüssel sind unterstrichen, Fremdschlüsseln ist ein <math>\uparrow</math> vorangestellt</p>	2	2	
c1	Es handelt sich um einen Join der Tabellen Vorstellung und Film über das Attribut FilmID. Selektiert wird nach dem Datum 10.4.2004. Projiziert werden die Attribute Datum, Uhrzeit, Titel und FSK und zwar geordnet nach der Uhrzeit. Ausgabe des Filmprogramms für den 10.4.2004	3		
c2	Geschachtelte Select-Anweisung. Die innere Select-Anweisung ermittelt mit einem Join über die FilmNr und einer Selektion die Action-Filme und bestimmt mittels MAX(Verkauft) den Action-Film mit größter Besucherzahl. Die äußere Select-Anweisung gibt dann einige Daten dieses Filmes aus. Bestimmung des Actionfilms mit größter Besucherzahl in einer Vorstellung.		3	
d1	<pre>SELECT FilmID, Titel, Datum, Uhrzeit FROM Vorstellung, Film WHERE     Film.FilmID = Vorstellung.FilmID AND     Datum Between #1/1/504# AND #5/31/2004# AND     Kategorie='Abenteuer';</pre>	3		
d2	<pre>SELECT VorstellungID, Datum, Uhrzeit</pre>		3	

	erwartete Schülerleistung	I	II	III
d3	<p>FROM Vorstellung, Saal, Film WHERE Saal.Saalnummer = Vorstellung.Saalnummer AND Vorstellung.FilmID = Film.FilmID AND Titel = 'Herr der Ringe III' AND Verkauft &lt; Sitzplätze</p> <p>SELECT Film.FilmID, Sum(Verkauft) FROM Film, Vorstellung WHERE Film.FilmID=Vorstellung.FilmID GROUP BY Film.FilmID HAVING Sum(Verkauft) &gt; 1000</p>			3
e	<p>Das ER-Diagramm muss um eine Tabelle Reservierung erweitert werden, die in einer n:1-Beziehung zur Tabelle Vorstellung steht. Technische Aspekte sind z. B. Transaktionsverwaltung, Datensicherheit und Web-Datenbanken Gesellschaftliche Aspekte sind z. B. Datenschutz und Vergleich zwischen telefonischer und online-Reservierung</p>		3	3
	<b>Σ 35 BE</b>	11	18	6

### ER-Diagramm zu Aufgabe a



## B1 Geldbeträge

### Aufgabenart

Diese Aufgabe behandelt die Themen Grammatiken und formale Sprachen sowie endliche Automaten und die Umsetzung eines Akzeptors mit der Programmiersprache Prolog.

### Voraussetzungen des Lehrplans

Zur Lösung der Aufgabe werden folgende verbindlichen Inhalte des Lehrplans benötigt: Formale Sprachen und Grammatiken samt Anwendung mit Syntaxdiagrammen, Zustandsdiagramm und Simulation eines endlichen Automaten, Kenntnisse in Prolog.

### Bewertungseinheiten

	erwartete Schülerleistung	I	II	III
a	<p>Das Syntaxdiagramm für Betrag lässt den Betrag 0 EUR und positive EUR-Beträge mit maximal drei Stellen zu. Beträge mit führenden Nullen sind ausgeschlossen. Es ist die Sprache der ganzzahligen Euro-Beträge von 0 EUR bis 999 EUR.</p>	3	3	
b	<p>Das Zustandsdiagramm sollte gemäß der Struktur der Syntaxdiagramme aufgebaut werden. Es sind nicht deterministische endliche Automaten mit oder ohne Epsilon-Übergängen möglich.</p>	2	4	
c	<p>Die Zustände des Automaten sind Variablen in der Grammatik. Jeder Übergang entspricht einer Produktion. Abkürzende Schreibweisen für die Behandlung der Ziffern 0 bis 9 sind erlaubt.</p> <p> <math>S \rightarrow 0V</math>  <math>S \rightarrow 1V \mid 2V \mid \dots \mid 9V</math>  <math>S \rightarrow 1P_1 \mid 2P_1 \mid \dots \mid 9P_1</math>  <math>S \rightarrow 1P_2 \mid 2P_2 \mid \dots \mid 9P_2</math>  <math>P_1 \rightarrow 0P_2 \mid 1P_2 \mid \dots \mid 9P_2</math>  <math>P_2 \rightarrow 0V \mid 1V \mid \dots \mid 9V</math>  <math>V \rightarrow " "W_1</math> </p>		5	

	erwartete Schülerleistung	I	II	III
	$V \rightarrow , K_1$ $K_1 \rightarrow 0K_2 \mid 1K_2 \mid \dots \mid 9K_2$ $K_2 \rightarrow 0K_3 \mid 1K_3 \mid \dots \mid 9K_3$ $K_3 \rightarrow "W_1$ $W_1 \rightarrow EW_2$ $W_2 \rightarrow UW_3$ $W_3 \rightarrow R$ Die Ableitung muss schrittweise stattfinden und der Bezug zur vorher erstellten Grammatik vorhanden sein.  $S \rightarrow 5P_1 \rightarrow 54P_2 \rightarrow 546V \rightarrow 546,K_1 \rightarrow 546,8K_2 \rightarrow 546,80K_3 \rightarrow 546,80 W_1 \rightarrow 546,80 EW_2 \rightarrow 546,80 EUW_3 \rightarrow 546,80 EUR$	2		
d	Es handelt sich um einen Akzeptor für das Syntaxdiagramm Positiv. Die Zahlen 0 bis 999 werden akzeptiert. Die Bedeutung der Parameter Ein und Aus muss klar werden: der Parameter Ein ist die Eingabeliste, der Parameter Aus die Liste der noch nicht verarbeiteten Zeichen. akzeptiere_zp akzeptiert obligatorisch eine der Ziffern 1 bis 9, akzeptiere_zn akzeptiert optional eine der Ziffern 0 bis 9.		4	
e	Es ist ein vollständiger Akzeptor für Geldbeträge in Prolog zu entwickeln (wobei Aufgabe e zu berücksichtigen ist). akzeptiere_betrag(Betrag):- akzeptiere_null_positiv(Betrag, Rest), akzeptiere_komma(Rest, Rest1), akzeptiere_eur(Rest1).  akzeptiere_null_positiv(['0' R], R). akzeptiere_null_positiv(L1, L2):- akzeptiere_positiv(L1, L2).  akzeptiere_komma(L1, L2):- L1 = [' ', R1], akzeptiere_zk(R1, R2), akzeptiere_zk(R2, L2). akzeptiere_komma(L, L).  akzeptiere_zk([K R], R):- member(K, ['0','1','2','3','4','5','6','7','8','9']).  akzeptiere_eur([' ', 'E', 'U', 'R']).		2	5
	<b>Σ 30 BE</b>	10	15	5



## B 2 Kellerautomat

### Aufgabenart

Die Aufgabe gehört zum Halbjahresthema 13.1 „Konzepte und Anwendungen der theoretischen Informatik“. Die Sprache kann leicht anhand der Beispiele erkannt werden. In Aufgabe b wird ein deterministischer Kellerautomat erwartet. Die Darstellung des Kellerautomaten kann durch Angabe der Übergangsfunktion oder in einem Zustandsdiagramm erfolgen.

### Voraussetzungen des Lehrplans

Zur Lösung der Aufgabe werden folgende verbindlichen Inhalte des Lehrplans benötigt: Formale Sprachen und Grammatiken, Endliche Automaten, Simulation realer Automaten, Kellerautomaten..

### Bewertungseinheiten

erwartete Schülerleistung		I	II	III																																																																														
a	<p>Einige Worte der Sprache L: aabcccbbaa, aca, bbbccbbb, c, aabbbccbbbaa</p> <p>Beschreibung: Die Worte sind spiegelsymmetrisch aufgebaut (Palindrome). Im Mittelteil muss mindestens ein c stehen. Der linke Teil beginnt mit einer Anzahl m des Zeichens a, dann folgt eine Anzahl n des Zeichens b. Die Anzahlen m und n können auch 0 sein. Der rechte Teil des Wortes ist spiegelsymmetrisch zum linken.</p> <p>Begründung: Um den linken und rechten Teil eines Wortes zu vergleichen, muss die Information über die Anzahlen m und n aufbewahrt werden. Da Information nur in den Zuständen eines endlichen Automaten gespeichert werden kann, braucht man mindestens m Zustände um sich die Anzahl m zu merken. Weil m beliebig groß werden kann, kommt man mit einer endlichen Anzahl von Zuständen nicht aus. Daher kann es keinen endlichen Automaten geben, der diese Sprache erkennt.</p> <p>Die Sprache ist kontextfrei und daher vom Chomsky-Typ 2.</p>	2  2      1		2																																																																														
b	<p>Der Kellerautomat wird dargestellt durch:</p> <p>Eingabezeichen = {a, b, c, ε}, Kellerzeichen = {A, B, C, ε, #}, # = Kellerboden, Zustände = {p, q, r, s, t, z}, s = Startzustand, z = Endzustand und die Übergangsfunktion. Die Übergangsfunktion kann durch Angabe der einzelnen Übergänge oder als Zustandsdiagramm dargestellt werden. Die Übergänge haben die Form:</p> <table><tr><td>Zustand</td><td>Eingabezeichen</td><td>oberstes Kellerzeichen</td><td>→</td><td>Folgezustand</td><td>String, durch den das oberste Kellerzeichen ersetzt wird</td></tr><tr><td>s a #</td><td>→</td><td>s A #</td><td></td><td></td><td></td></tr><tr><td>s b #</td><td>→</td><td>p B #</td><td></td><td></td><td></td></tr><tr><td>s c #</td><td>→</td><td>q #</td><td></td><td></td><td></td></tr><tr><td>r b B</td><td>→</td><td>r ε</td><td></td><td></td><td></td></tr><tr><td>q ε #</td><td>→</td><td>z #</td><td></td><td></td><td></td></tr></table> <p>Beim Start wird davon ausgegangen, dass im Keller nur das Kellerzeichen # ist und das Eingabezeichen das erste Zeichen des zu prüfenden Wortes ist.</p> <table><tr><td>s a A</td><td>→</td><td>s A A</td><td></td><td></td><td></td></tr><tr><td>p b B</td><td>→</td><td>p B B</td><td></td><td></td><td></td></tr><tr><td>q c B</td><td>→</td><td>q B</td><td></td><td></td><td></td></tr><tr><td>q c A</td><td>→</td><td>q A</td><td></td><td></td><td></td></tr><tr><td>q b B</td><td>→</td><td>r ε</td><td></td><td></td><td></td></tr><tr><td>t a A</td><td>→</td><td>t ε</td><td></td><td></td><td></td></tr><tr><td>t ε #</td><td>→</td><td>z #</td><td></td><td></td><td></td></tr></table>	Zustand	Eingabezeichen	oberstes Kellerzeichen	→	Folgezustand	String, durch den das oberste Kellerzeichen ersetzt wird	s a #	→	s A #				s b #	→	p B #				s c #	→	q #				r b B	→	r ε				q ε #	→	z #				s a A	→	s A A				p b B	→	p B B				q c B	→	q B				q c A	→	q A				q b B	→	r ε				t a A	→	t ε				t ε #	→	z #					15	
Zustand	Eingabezeichen	oberstes Kellerzeichen	→	Folgezustand	String, durch den das oberste Kellerzeichen ersetzt wird																																																																													
s a #	→	s A #																																																																																
s b #	→	p B #																																																																																
s c #	→	q #																																																																																
r b B	→	r ε																																																																																
q ε #	→	z #																																																																																
s a A	→	s A A																																																																																
p b B	→	p B B																																																																																
q c B	→	q B																																																																																
q c A	→	q A																																																																																
q b B	→	r ε																																																																																
t a A	→	t ε																																																																																
t ε #	→	z #																																																																																

	erwartete Schülerleistung	I	II	III
	Alternative Lösung durch ein Zustandsdiagramm (siehe unten): Fehlerzustände werden nicht dargestellt. push x = Ablegen von x auf dem Keller pop = Entfernen des obersten Kellerelements nop = keine Kelleroperation			
c	Es werden lineare Strukturen benötigt für den Keller und das Eingabewort. An einem Ende der Struktur muss zugegriffen werden, um ein Zeichen anzufügen (push) oder zu löschen (pop). Zweckmäßig ist es auch, das erste Zeichen nur zu lesen (top) ohne es zu entfernen, da hiermit Fehlerzustände erkannt werden können.  Mögliche Realisierungen: 1. Struktur: lineare Liste Zugriffsmethoden: Anfügen, Löschen, Lesen des ersten Elementes 2. Struktur: array Zugriff direkt auf die Komponente, Verwalten des Index des ersten Elementes 3. Struktur: String Zugriff mit Stringoperationen am Anfang des Strings	5		
d	Ein Kellerautomat kann Information, z. B. eine Anzahl, speichern, in dem er entsprechend viele Symbole auf den Keller legt. Wird die Information benutzt, so wird sie zerstört, da die Symbole vom Keller abgeräumt werden. Bei der Sprache T kann im ersten Teil des Wortes zwar festgestellt werden, ob die Zahl der a und b gleich ist, aber diese Zahl ist dann bei der Prüfung des rechten Teils des Wortes nicht mehr vorhanden. Also kann nicht festgestellt werden, ob im rechten und linken Teil gleich viele a vorhanden sind.			3
	<b>Σ 30 BE</b>	10	15	5

### Zustandsdiagramm zu Aufgabe b

