

Lösungshinweise: Nur für die Hand der Lehrperson

Schriftliche Abiturprüfung 2014

Fach: **Informatik**

Kurstyp: G-Kurs

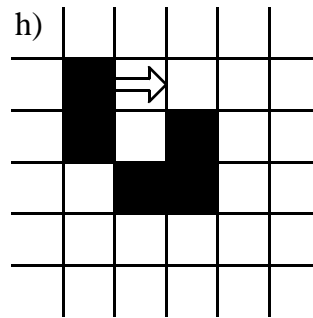
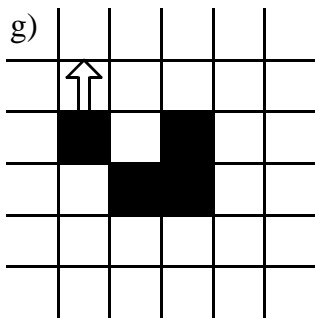
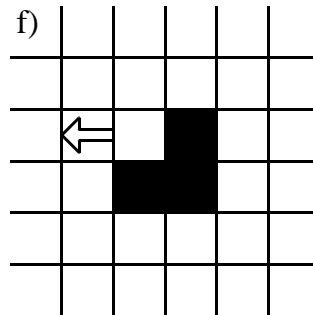
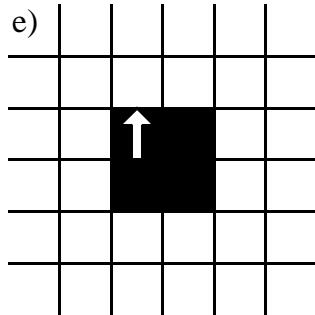
Bearbeitungszeit: 3 Zeitstunden

Hilfsmittel: nicht-programmierbarer Taschenrechner

Seitenzahl: Die Lösungshinweise umfassen mit Deckblatt, Punkteverteilungstabelle und Bewertungstabelle 10 Seiten.

1. Aufgabe (20 Punkte)

1.1 (4 Punkte) (Lehrplan: "Objektorientiertes Modellieren und Programmieren")



Die Abbildungen e) bis h) zeigen die vier nächsten Schritte.

1.2 (4 Punkte) (Lehrplan: "Objektorientiertes Modellieren und Programmieren")

Java-Code:

```
public class TPosition
{
    private int x;
    private int y;

    public TPosition(int x, int y)
    {
        this.x = x;
        this.y = y;
    }

    public int gibX()
    {
        return x;
    }

    public int gibY()
    {
        return y;
    }
}
```

PASCAL/Delphi-Code:

```
type TPosition = class(TObject)
private
    x: integer;
    y: integer;

public
    constructor erzeugen(xPos,yPos: integer);
    function gibX(): integer;
    function gibY(): integer;
end;

implementation
constructor TPosition.erzeugen(xPos,yPos: integer);
begin
    x := xPos;
    y := yPos;
end;

function TPosition.gibX(): integer;
begin
    result := x;
end;

function TPosition.gibY(): integer;
begin
    result := y;
end;
```

1.3 (5 Punkte) (Lehrplan: "Objektorientiertes Modellieren und Programmieren")**Java-Code:**

```
public TPosition fuehreSchritteAus(int anzahl)
{
    for (int i = 0; i < anzahl; i++)
    {
        TPosition pos = ameise.gibAktuellePosition();
        boolean schwarz = spielfeld.istSchwarz(pos);
        ameise.bewege(schwarz);
        spielfeld.faeerbeUm(pos);
    }
    return ameise.gibAktuellePosition();
}
```

PASCAL/Delphi-Code:

```
function TAutomat.fuehreSchritteAus(anzahl: integer): TPosition;
var i: integer;
pos: TPosition;
schwarz: boolean;

begin
    for i := 1 to anzahl do
        begin
            pos := ameise.gibAktuellePosition();
            schwarz := spielfeld.istSchwarz(pos);
            ameise.bewege(schwarz);
            spielfeld.faeerbeUm(pos);
        end;
    result := ameise.gibAktuellePosition();
end;
```

1.4 (5 Punkte) (Lehrplan: "Objektorientiertes Modellieren und Programmieren")

Java-Code:

```
public boolean istSchwarz(TPosition pos)
{
    return feld[pos.gibX()][pos.gibY()];
}

public void faerbeUm(TPosition pos)
{
    boolean schwarz = feld[pos.gibX()][pos.gibY()];
    feld[pos.gibX()][pos.gibY()] = !schwarz;
}
```

PASCAL/Delphi-Code:

```
function TSpielfeld.istSchwarz(pos: TPosition): boolean;
begin
    result := feld[pos.gibX()][pos.gibY()];
end;

procedure TSpielfeld.faelbeUm(pos: TPosition);
var schwarz: boolean;
begin
    schwarz := feld[pos.gibX()][pos.gibY()];
    feld[pos.gibX()][pos.gibY()] := not(schwarz);
end;
```

1.5 (2 Punkte) (Lehrplan: "Datentypen und Datenstrukturen")

Das Problem besteht darin, dass ein festgelegtes zweidimensionales Feld nicht ausreicht, um eine große Anzahl von Ausführungsschritten zu verwalten, und dass die Ameise früher oder später den Rand des Feldes erreichen wird. Um dieses Problem zu umgehen, könnte man die Positionen der schwarzen Zellen in einer linearen Liste verwalten.

Alternativ könnte das Feld auch bei jedem Erreichen des Randes in ein größeres Feld kopiert werden.

2. Aufgabe (20 Punkte)

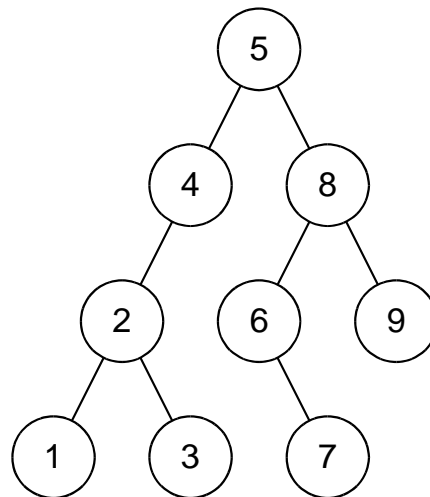
2.1 (13 Punkte) (Lehrplan: "Datentypen und Datenstrukturen")

2.1.1 (2 Punkte)

Der Baum ist genau dann ein binärer Suchbaum, wenn er leer ist oder wenn

- alle Elemente im linken Teilbaum kleiner als die Wurzel sind und
- alle Elemente im rechten Teilbaum größer als die Wurzel sind und
- linker und rechter Teilbaum binäre Suchbäume sind.

2.1.2 (4 Punkte)

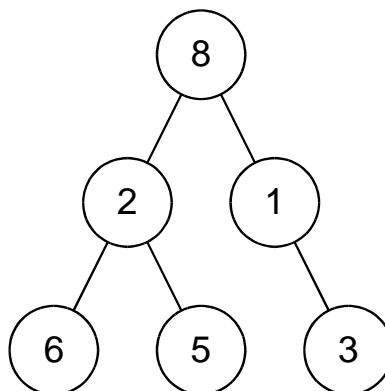


2.1.3 (2 Punkte)

Preorder-Durchlauf: 5, 4, 2, 1, 3, 8, 6, 7, 9

Inorder-Durchlauf: 1, 2, 3, 4, 5, 6, 7, 8, 9

2.1.4 (4 Punkte)



2.1.5 (1 Punkt)

Die Art des Durchlaufs bewirkt, dass in der Sequenz zunächst alle Zahlen auftreten, die kleiner als die Wurzel sind. Es folgen die Wurzel sowie alle größeren Zahlen.

Insofern steht die Wurzel in der Sequenz an der richtigen Position.

Nach demselben Argument treten auch die Zahlen des linken und rechten Teilbaums in der richtigen Reihenfolge auf.

2.2 (7 Punkte) (Lehrplan: "Funktionsweise von Computersystemen")**2.2.1 (2 Punkte)**

Fetch-Phase	Befehlscode wird in IR geladen. PC gibt an, welcher Befehl zu laden ist und wird am Ende dieser Phase inkrementiert.
Decode-Phase	Geladener Befehl wird decodiert, d.h. die Funktionseinheiten erhalten die erforderlichen Steuersignale.
Load-Phase	Operanden, die zur Befehlsausführung notwendig sind, werden geladen. Bei Befehlen ohne Operanden ist diese Phase leer.
Execute-Phase	Nach dem Laden aller Operanden wird der Befehl ausgeführt.

2.2.2 (5 Punkte)

```
0 JMP 4
1 DEF 0 ;Variable a
2 DEF 0 ;Variable b
3 DEF 0 ;Variable erg als Zähler für ganzzahlige Division

4 INM 1 ;a einlesen
5 INM 2 ;b einlesen

6 LDA 1
7 SUB 2 ;Subtrahiere b von a
8 JMS 14 ;Springe zur Ausgabe, wenn b einmal zu viel subtrahiert wurde
9 STA 1

10 LDA 3
11 INC ;Zähler erg wird um 1 erhöht
12 STA 3

13 JMP 6 ;Wiederhole die Subtraktion von b

14 OUT 3 ;Ausgabe ganzzahliges Ergebnis Division
15 OUT 1 ;Ausgabe ganzzahliger Rest
16 END
```

3. Aufgabe (20 Punkte)**3.1 (6 Punkte) (Lehrplan: "Automaten und formale Sprachen")****3.1.1 (3 Punkte)**

- a) Das Passwort entspricht den Regeln. (0,5 Punkte)
 b) Das Passwort entspricht nicht den Regeln, da mehr als zwei Buchstaben aufeinander folgen und das Passwort mit einer Ziffer endet. (1 Punkt)
 c) Das Passwort entspricht nicht den Regeln, da # nicht zu den angegebenen Sonderzeichen zählt. (1 Punkt)
 d) Das Passwort entspricht den Regeln. (0,5 Punkte)

3.1.2 (3 Punkte)

$S \rightarrow aB \mid \dots \mid zB$ //Beginn mit Kleinbuchstaben

$B \rightarrow aC \mid \dots \mid zC$ //gefolgt von Kleinbuchstaben

$B \rightarrow @ F \mid \dots \mid \% F$ //oder Sonderzeichen/Ziffern

$B \rightarrow 0F \mid \dots \mid 9F$

$C \rightarrow @ D \mid \dots \mid \% D$ //nach 2 Buchstaben muss Sonderzeichen

$C \rightarrow 0D \mid \dots \mid 9D$ //oder Ziffer folgen

$D \rightarrow a \mid \dots \mid z$ //Ende des Passwortes mit Kleinbuchstaben

$F \rightarrow aD \mid \dots \mid zD$ //nach Sonderzeichen/Ziffer kann entweder ein zweites

$F \rightarrow @ D \mid \dots \mid \% D$ //Sonderzeichen/Ziffer oder ein

$F \rightarrow 0D \mid \dots \mid 9D$ //Kleinbuchstabe stehen

3.2 (7 Punkte) (Lehrplan: "Automaten und formale Sprachen")**3.2.1 (2 Punkte)**

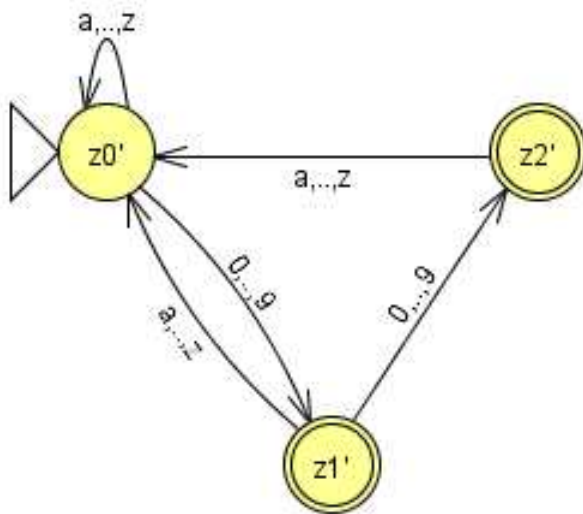
- In Passwörtern können nur Kleinbuchstaben und Ziffern vorkommen.
- Passwörter enden auf eine oder zwei Ziffern.

3.2.2 (5 Punkte)

Teilmengenkonstruktion:

$\bar{\emptyset}$	$0, \dots, 9$	a, \dots, z
$z_0' := \{z_0\}$	$\{z_1, z_2\}$	$\{z_0\}$
$z_1' := \{z_1, z_2\}$	$\{z_1\}$	$\{z_0\}$
$z_2' := \{z_1\}$	\emptyset	$\{z_0\}$

Übergangsgraph:



3.3 (7 Punkte) (Lehrplan: "Kryptographie")

3.3.1 (3 Punkte)

Schlüsseltabelle:

E	D	W	A	R
S	N	O	B	C
F	G	H	I/J	K
L	M	P	Q	T
U	V	X	Y	Z

Verschlüsselung:

Klartext:	IN	FO	RM	AT	IK
Geheimtext:	GB	HS	DT	RQ	KF

3.3.2 (2 Punkte)

Geheimtext:	PG	RS	IC	XS	PQ	GL	PO	QY
Klartext:	AB	HO	ER	SK	AN	DA	LN	SA

3.3.3 (2 Punkte)

- Buchstabenpaare des Klartextalphabets werden durch Buchstabenpaare des Geheimtextalphabets ersetzt. => Substitution
- Jedem Buchstabenpaar des Klartextalphabets ist ein eindeutiges Buchstabenpaar des Geheimtextalphabets zugeordnet.
=> monoalphabetische Substitution
- Kryptoanalyse durch Häufigkeitsanalyse

Punkteverteilungstabelle**Aufgabe 1:**

Teilaufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
1.1	4	4		
1.2	4		4	
1.3	5		5	
1.4	5		5	
1.5	2			2
	20	4	14	2
		20%	70%	10%

Aufgabe 2:

Teilaufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
2.1	13	2	7	4
2.2	7	2	5	
	20	4	12	4
		20%	60%	20%

Aufgabe 3:

Teilaufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
3.1	6		3	3
3.2	7		7	
3.3	7	2	5	
	20	2	15	3
		10%	75%	15%

Punkteverteilungstabelle (insgesamt):

Aufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
1	20	4	14	2
2	20	4	12	4
3	20	2	15	3
	60	10	41	9
		17%	68%	15%

Bewertungstabelle

Prozent der maximal erreichbaren Rohpunktzahl	Note	Punktzahl
ab 97% bis 100% der max. Punktzahl	sehr gut	15 P
ab 93% bis weniger als 97%		14 P
ab 90% bis weniger als 93%		13 P
ab 85% bis weniger als 90%	gut	12 P
ab 80% bis weniger als 85%		11 P
ab 75% bis weniger als 80%		10 P
ab 70% bis weniger als 75%	befriedigend	09 P
ab 65% bis weniger als 70%		08 P
ab 60% bis weniger als 65%		07 P
ab 55% bis weniger als 60%	ausreichend	06 P
ab 50% bis weniger als 55%		05 P
ab 45% bis weniger als 50%		04 P
ab 38% bis weniger als 45%	mangelhaft	03 P
ab 32% bis weniger als 38%		02 P
ab 25% bis weniger als 32%		01 P
weniger als 25% der max. Punktzahl	ungenügend	00 P

- Ende der Lösungshinweise -