Mecklenburg-Vorpommern

Das nachstehende Prüfungsdokument soll den Schülerinnen und Schülern des Landes sowohl zur individuellen Prüfungsvorbereitung als auch im Rahmen des Unterrichts in Lernsituationen zur Verfügung gestellt werden.

Abitur 2025

Informatik

Grundkurs

Prüfungsaufgaben und Musterlösungen

1 Pflichtaufgabe: Ticketdatenbank

Im Informatikunterricht wurde eine Ticketdatenbank modelliert, die Informationen über Veranstaltungen, Ticketverkäufe und Kundendaten verwalten soll.

Die Abbildung 1 zeigt einen Entwurf des Datenbanksystems und die Abbildung 2 ein Relationenschema.

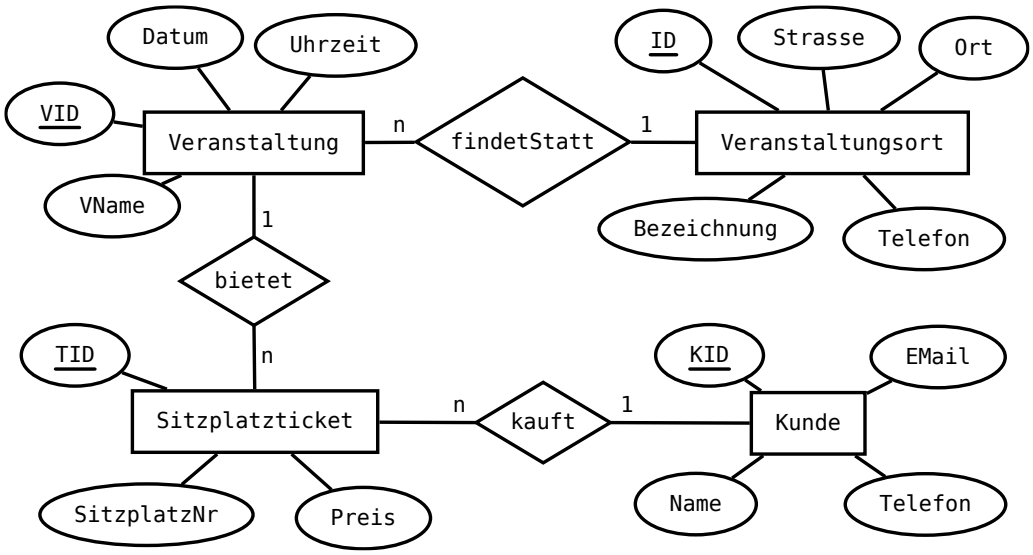


Abbildung 1

```
Kunde(KID, Name, Telefon, EMail)
Sitzplatzticket(TID, ↑KID, Preis, SitzplatzNr)
```

Abbildung 2

1.1 Modellierung

1.1.1 Ordnen Sie jeder Entwurfsetappe der Datenbankentwicklung deren Ergebnis zu. 3 BE

1.1.2 Interpretieren Sie den Beziehungstyp `bietet` in der Abbildung 1. 2 BE

1.1.3 Begründen Sie, dass der Sachverhalt `Kunde-kauft-Sitzplatzticket` in der Abbildung 1 korrekt in das Relationenschema in der Abbildung 2 überführt wurde. 2 BE

1.2 Arbeit mit dem Datenbanksystem

Der folgende Sachverhalt soll im Datenbanksystem gespeichert werden: Herr Schnell kauft das Ticket mit der ID 1 für die Sitzplatznummer 10 zu 45,50 EUR. Er besitzt die ID 102, die Telefonnummer 053/1678 und die E-Mail-Adresse kontakt@schnell.de.

1.2.1 Dem Attribut `Telefon` in der Abbildung 2 wird der Datentyp `Text` zugeordnet. 1 BE
Begründen Sie die Eignung des Datentyps für den Sachverhalt.

1.2.2 Implementieren Sie das Relationenschema aus der Abbildung 2 in ein neues Datenbanksystem. 7 BE
Wählen Sie für die Attribute passende Datentypen.
Legen Sie die Primär- und Fremdschlüssel an.
Ergänzen Sie den oben genannten Sachverhalt im Datenbanksystem.

Hinweis: Bei Implementierungsproblemen sind diese und das geplante Vorgehen unter Nutzung der Anlage 1 zu beschreiben.

1.2.3 Formulieren Sie die Abfragen für das Relationenschema aus Abbildung 2 in SQL. 7 BE

Abfrage 1
Angabe der Sitzplatznummern, bei denen ein Sitzplatzticket mindestens 20 EUR aber maximal 50 EUR kostet

Abfrage 2
Angabe aller Informationen zum Sitzplatzticket mit der höchsten Sitzplatznummer

Abfrage 3
Angabe der Kundennamen und Sitzplatznummern gekaufter Tickets von Kunden, die eine Mailadresse besitzen, die die Zeichenkette `abipost` enthält

1.3 Onlineportal

Ticketverkauf und Veranstaltungswerbung werden über ein Portal im Internet realisiert. Im Ordner *Aufgabe 1* liegt ein Kommunikationsmitschnitt in der Datei *mitschnitt.ods* vor, der den Zugriff eines Kunden auf den Ticketserver protokolliert.

1.3.1 Geben Sie die IP-Adressen des Kundenrechners und des Ticketservers an. 2 BE

1.3.2 Im gegebenen Anwendungsfall wird das Transportprotokoll TCP verwendet. Beschreiben Sie einen Vorteil von TCP im Vergleich zum Transportprotokoll UDP. 1 BE

1.3.3 Ticketbesitzer können Videos ihrer besuchten Veranstaltungen vom Server abrufen. Berechnen Sie die minimale Downloadzeit eines Videos mit einer Größe von 1,5 GByte bei einer Übertragungsrate von 16 MBit/s. 2 BE

1.3.4 Beim Kauf von Tickets im Portal sind Maßnahmen zum Datenschutz notwendig. Erläutern Sie zwei Maßnahmen. 3 BE

2 Wahlaufgabe: Kartenspiel

Im Rahmen eines Programmierprojekts soll ein Kartenspiel entwickelt werden. Der Ordner *Aufgabe 2* enthält eine Teilimplementierung des Softwareprojekts. Diese ermöglicht die Verwaltung eines Aufnahmestapels, eines Ablagestapels und von Handkarten eines Spielers. Die Abbildung 3 zeigt das zugrundeliegende Klassendiagramm.

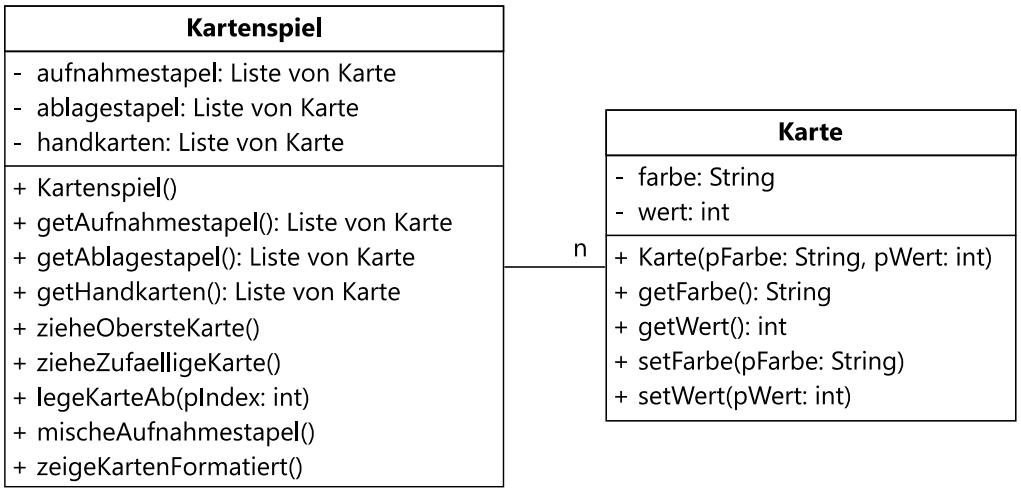


Abbildung 3

2.1 Konzepte der Objektorientierung

2.1.1 Interpretieren Sie die Beziehung zwischen den Klassen `Kartenspiel` und `Karte`. Beschreiben Sie die Umsetzung der Beziehung. 3 BE

2.1.2 Ordnen Sie den Methodenarten „Methode mit Parameter“ und „Methode ohne Rückgabe“ je eine Methode aus der Klasse `Kartenspiel` zu. 2 BE

2.1.3 Erläutern Sie das Konzept der Datenkapselung anhand der Klasse `Karte`. 4 BE

2.1.4 Das Projekt erfüllt das Softwarequalitätskriterium Wiederverwendbarkeit. Begründen Sie diese Aussage an zwei unterschiedlichen Beispielen. 2 BE

2.2 Weiterentwicklung der Klasse Kartenspiel

2.2.1 Geben Sie mithilfe des Konstruktors der Klasse `Kartenspiel` die Anzahl sowie Farbe und Wert der im Spiel erzeugten Karten an. 2 BE

2.2.2 Ergänzen Sie in der Methode `zieheZufaelligeKarte` für die angegebenen Schritte geeignete Kommentare, um deren Arbeitsweise zu beschreiben. 3 BE

2.2.3 Erweitern Sie die Methode `mischeAufnahmestapel` zum Mischen des Aufnahmestapels entsprechend dem Algorithmus in der Abbildung 4. 4 BE

Setze anzahl auf Anzahl der Elemente des Aufnahmestapels
FÜR i = 0 BIS (anzahl – 1)
Setze indexZufall auf ganzzahlige Zufallszahl von 0 bis anzahl - 1
Setze karte auf Objekt von aufnahmestapel mit Index i
Setze karteZufall auf Objekt von aufnahmestapel mit Index indexZufall
Ersetze in aufnahmestapel das Element mit Index i durch karteZufall
Ersetze in aufnahmestapel das Element mit Index indexZufall durch karte

Abbildung 4

2.2.4 Die Methode `legeKarteAb` soll es ermöglichen, eine über ihren Index ausgewählte Karte aus den Handkarten auf den Ablagestapel abzulegen. 3 BE
Ergänzen Sie die Methode um diese Funktionalität.

2.3 Kartenprüfgeräte

Kartenprüfgeräte basieren auf dem Modell eines von-Neumann-Rechners und prüfen Kartensets auf Fehlerfreiheit. Im Rahmen eines Qualitätstests wird jeder gescannten Karte ein Qualitätswert von 1 bis 10 zugewiesen. Ein Teil des Prüfalgorithmus liegt in der Datei *a2.ram* im Ordner *Aufgabe 2* für das Simulationsprogramm *Johnny* vor. In den Speicherzellen 0 bis 6 befindet sich der selbstveränderliche Algorithmus. Die gemessenen Qualitätswerte beginnen ab Zelle 11. Das Ergebnis des Algorithmus wird in Zelle 10 gespeichert. Die *Anlage 2* zeigt den Quelltext des Algorithmus.

2.3.1 Nennen Sie zwei Arbeitsprinzipien eines von-Neumann-Rechners. 2 BE

2.3.2 Erläutern Sie die Befehle in den Speicherzellen 0 und 1. 3 BE

2.3.3 Ermitteln Sie den Zweck des Algorithmus. 2 BE

3 Wahlaufgabe: Strichcodes

Supermärkte nutzen Strichcodes zur Identifizierung ihrer Artikel. Lesegeräte erfassen die Folge der sich abwechselnden schwarzen und weißen Striche und ermitteln den Strichcode.

Ein schulisches Programmierprojekt nutzt dieses Prinzip. Hier haben die Striche eine Breite von 1, 2 oder 3 Einheiten. Jede Folge beginnt und endet stets mit einem schwarzen Strich, wobei ein einzelner schwarzer Strich bereits eine korrekte Darstellung ist. Die Striche werden von links nach rechts gelesen und als Zeichenfolge codiert. Für jeden Strich wird der Farbwert (`s` oder `w`) und die Breite angegeben. Die Abbildung 5 zeigt einen Strichcode mit Angabe der Breite der Striche. Seine Zeichenfolge lautet `s2w1s2w3s1w2s2w1s3`.

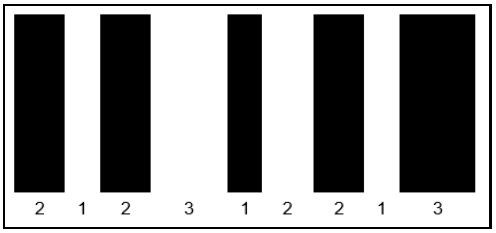


Abbildung 5

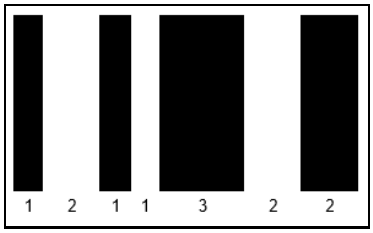


Abbildung 6

3.1 Strichcode und Automat

Die *Anlage 3* zeigt die Überführungsfunktion eines Automaten zur Prüfung von Zeichenfolgen. Der Automat akzeptiert auch einige nicht korrekte Zeichenfolgen.

3.1.1 Geben Sie den Strichcode aus der Abbildung 6 als Zeichenfolge an. 2 BE
Stellen Sie den Strichcode zu der Zeichenfolge `s2w2s1w2s3` grafisch dar.

3.1.2 Begründen Sie, dass `s2w1w2s3` und `s2w2` keine korrekten Zeichenfolgen sind. 2 BE

3.1.3 Zeigen Sie durch Angabe der durchlaufenen Zustände, dass der Automat die Zeichenfolgen aus der Aufgabe 3.1.2 akzeptiert. 2 BE

3.1.4 Modifizieren Sie die Überführungsfunktion des Automaten, sodass nur korrekte Zeichenfolgen akzeptiert werden. 3 BE

3.2 Softwareprojekt

Darstellungen von Strichcodes, die von rechts nach links gelesen wurden, sollen auch zulässig sein. Der Ordner *Aufgabe 3* enthält ein unvollständiges Softwareprojekt, das Zeichenfolgen verarbeitet.

3.2.1 Erläutern Sie die Aufgaben des Konstruktors der Klasse `Codemanipulation`. 3 BE

3.2.2 Protokollieren Sie die Arbeitsweise der Methode `erweitern` der Klasse `Codemanipulation` für `strichcode` = "`s2w2s1w2s3`". 4 BE
Beschreiben Sie allgemein die Veränderung der Zeichenfolge eines gegebenen Strichcodes durch diese Methode.

3.2.3 Ein Schüler A schlägt folgende alternative Codierung der Zeichenfolge vor: 4 BE
Für jeden Strich wird der Farbwert (`s` oder `w`) angegeben. Die Strichbreite wird durch die Anzahl der Farbwerte dargestellt. Der Strichcode in Abbildung 5 wird somit durch die Zeichenfolge `sswssswwwswwsswsss` dargestellt.
Ergänzen Sie die Methode `transformieren` zur Umsetzung der alternativen Codierung entsprechend dem Algorithmus in Abbildung 7.

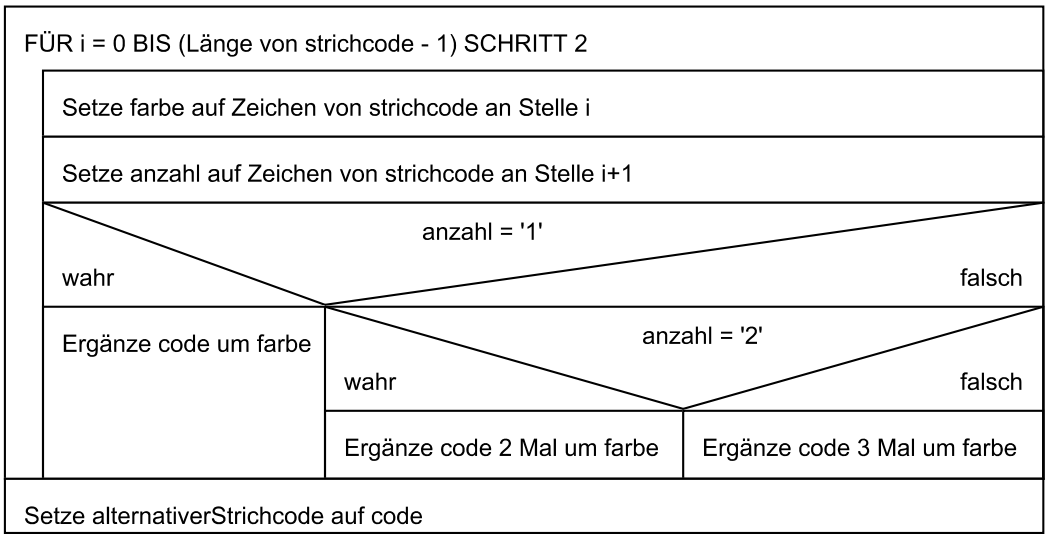


Abbildung 7

3.2.4 Ein Schüler B hat die Idee, die abwechselnden schwarzen und weißen Striche ausschließlich durch Angabe der Strichbreiten (`1`, `2` oder `3`) zu codieren. Der Strichcode in Abbildung 5 wird dann durch die Zeichenfolge `212312213` dargestellt. 4 BE
Beschreiben Sie Vor- und Nachteile der Alternativen von Schüler A und Schüler B im Vergleich zur ursprünglichen Codierung.

3.3 Datenübertragung

In einem Supermarkt können Kunden den Strichcode von Artikeln an einem Gerät erfassen und erhalten dann weitere Informationen. Das Gerät kommuniziert über das Internet mit einem Server, der über die IP-Adresse 87.142.80.212/16 erreichbar ist.

3.3.1 Beschreiben Sie die Aufgaben von Routern und Servern in vernetzten Systemen. 3 BE
Ordnen Sie den Router in ein Schichtenmodell ein.

3.3.2 Geben Sie die als Suffix gegebene Netzmaske in Dezimal-Punkt-Notation an. 3 BE
Erläutern Sie den Einfluss der Netzmaske auf die Interpretation der IP-Adresse des Servers.

1 Pflichtaufgabe: Ticketdatenbank

<table>
<tr><th>1.1</th><th>Modellierung</th><th>I</th><th>II</th><th>III</th></tr>
<tr><td>1.1.1</td><td>konzeptioneller Entwurf → ER-Modell
logischer Entwurf → Relationenschema
physischer Entwurf → Datenbanksystem</td><td>3</td><td>0</td><td>0</td></tr>
<tr><td>1.1.2</td><td>Eine Veranstaltung bietet mehrere Sitzplatztickets an.
Ein Sitzplatzticket gehört zu einer Veranstaltung.</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>1.1.3</td><td>Jeder Entitätstyp mit seinen Attributen wurde in eine eigenständige Tabelle überführt. Die Primärschlüssel wurden gekennzeichnet.
Der Beziehungstyp <code>kauft</code> wurde aufgrund der Kardinalität 1:n nicht in eine eigenständige Tabelle überführt. Stattdessen wurde <code>KID</code> als Fremdschlüssel in der Tabelle <code>Sitzplatzticket</code> festgelegt.</td><td>1</td><td>1</td><td>0</td></tr>
<tr><th>1.2</th><th>Arbeit mit dem Datenbanksystem</th><th></th><th></th><th></th></tr>
<tr><td>1.2.1</td><td>Telefonnummern sind Folgen von Ziffern und evtl. anderen Zeichen wie „+“, „/“ oder Leerzeichen. Je nach Verwendungszweck ist ggf. eine führende Null abzubilden. Nur mit dem Datentyp <code>Text</code> kann dies realisiert werden.</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1.2.2</td><td>Erzeugen einer neuen Datenbankdatei
Erzeugen der Tabelle <code>Kunde</code> mit <code>KID</code> als Primärschlüssel vom Typ INTEGER und den weiteren Attributen vom Typ TEXT
Erzeugen der Tabelle <code>Sitzplatzticket</code> mit <code>TID</code> als Primärschlüssel vom Typ INTEGER, des Attributs <code>KID</code> vom Typ INTEGER als Fremdschlüssel auf das Attribut <code>KID</code> der Tabelle <code>Kunde</code> und der weiteren Attribute mit passendem Datentyp
Erfassen der Daten des Sachverhalts
Kunde
<table>
<tr><td><u>KID</u></td><td>Name</td><td>Telefon</td><td>EMail</td></tr>
<tr><td>102</td><td>Schnell</td><td>053/1678</td><td>kontakt@schnell.de</td></tr>
</table>
Sitzplatzticket
<table>
<tr><td><u>TID</u></td><td>↑KID</td><td>Preis</td><td>SitzplatzNr</td></tr>
<tr><td>1</td><td>102</td><td>45.50</td><td>10</td></tr>
</table></td><td>2</td><td>4</td><td>1</td></tr>
</table>

1.2.3	Abfrage 1
	`SELECT SitzplatzNr`
	`FROM Sitzplatzticket`
	`WHERE Preis BETWEEN 20 AND 50`
		1	1	0

	Abfrage 2
	`SELECT *`
	`FROM Sitzplatzticket`
	`ORDER BY SitzplatzNr DESC`
	`LIMIT 1`
		0	1	1

	Abfrage 3
	`SELECT Kunde.Name, Sitzplatzticket.SitzplatzNr`
	`FROM Sitzplatzticket INNER JOIN Kunde`
	`  ON Sitzplatzticket.KID = Kunde.KID`
	`WHERE Kunde.EMail LIKE '%abipost%'`
		0	1	2

1.3	**Onlineportal**			
1.3.1	IP-Adresse des Kundenrechners: 43.17.118.1 IP-Adresse des Ticketservers: 169.12.55.1	2	0	0
1.3.2	TCP baut im Vergleich zur Alternative UDP eine zuverlässige Verbindung mithilfe eines Handshakes auf und ist unter anderem in der Lage, auf den Verlust von Datenpaketen zu reagieren. Dies ist bei der Übertragung einer Internetseite sinnvoll.	0	1	0
1.3.3	$\frac{1{,}5\ \text{GByte}}{16\ \text{MBit/s}} = \frac{1500 \cdot 8\ \text{MBit}}{16\ \text{MBit}\cdot\text{s}^{-1}} = 750\ \text{s} = 12{,}5\ \text{min}$ *Hinweis: Die Umrechnung von GByte in MByte wurde entsprechend der Definition der Dezimalpräfixe mit 1000 durchgeführt. Sollte der Prüfling die Dezimalpräfixe als Binärpräfixe interpretiert und mit 1024 gerechnet haben, ist dies aufgrund der historischen Verwendung lediglich als Formverstoß zu bewerten.*	0	2	0
1.3.4	Maßnahmen mit geeignetem Beispielbezug: Einholung der Zustimmung des Kunden zur Speicherung personengebundener Daten durch Bestätigung der AGB, Erhebung ausschließlich zweckgebundener Daten beispielsweise Telefon als notwendiges Attribut für die schnelle Kontaktaufnahme bei Veranstaltungsabsage, Löschung der Daten nach Ablauf der Speicherfrist beispielweise nach der Veranstaltung, …	0	1	2
	gesamt	**9**	**15**	**6**

2 Wahlaufgabe: Kartenspiel

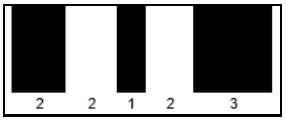
2.1	**Konzepte der Objektorientierung**	**I**	**II**	**III**
2.1.1	Assoziation. Ein Kartenspiel hat mehrere Karten. Umsetzung: Die Klasse `Kartenspiel` hat drei Attribute vom Typ `Kartenliste`. Jede Liste verwaltet mehrere Objekte der Klasse `Karte`.	2	1	0
2.1.2	Methode mit Parameter: `legeKarteAb` Methode ohne Rückgabe: bspw. `legeKarteAb` oder `zieheObersteKarte`	0	2	0
2.1.3	Die Datenkapselung beschreibt das Geheimnisprinzip der Objektorientierung und dient dazu, den direkten Zugriff auf die internen Daten und bestimmte Methoden einer Klasse zu beschränken. Die Umsetzung erfolgt in Java mit den Schlüsselwörtern `public` und `private`. `public` ermöglicht den Zugriff durch Objekte anderer Klassen. `private` verhindert den Zugriff durch Objekte anderer Klassen. In der Klasse sind die Attribute `farbe` und `wert` `private` deklariert und somit nicht aus anderen Klassen direkt erreichbar. Alle Methoden der Klasse sind öffentlich.	3	1	0
2.1.4	Die Klasse `Karte` kann als Grundlage für andere Spiele mit Karten dienen, da Karten in der Regel durch ihre Farbe und ihren Wert beschrieben werden. Die Methode `mischeAufnahmestapel` der Klasse `Kartenspiel` ist in anderen Kontexten wiederverwendbar, sobald Listenelemente zufällig angeordnet werden sollen.	0	1	1
2.2	**Weiterentwicklung der Klasse Kartenspiel**			
2.2.1	Es werden 10 rote und 10 blaue Karten jeweils mit den Werten von 1 bis 10 erzeugt.	0	2	0
2.2.2	Schritt 1: Erzeugung der Variablen `anzahl` und Belegung mit der Anzahl der Elemente des Aufnahmestapels Schritt 2: Belegung der Variablen `i` mit einer ganzzahligen Zufallszahl im Bereich von `0` bis `anzahl-1` Schritt 3: Entfernen der `i`-ten Karte aus dem Aufnahmestapel und Anfügen an die Liste `handkarten`	1	2	0
2.2.3	1. Umsetzung der Zählschleife 2. Festlegung des Zufallsindex 3. Extrahieren der Elemente 4. Ersetzen der Elemente	0	3	1

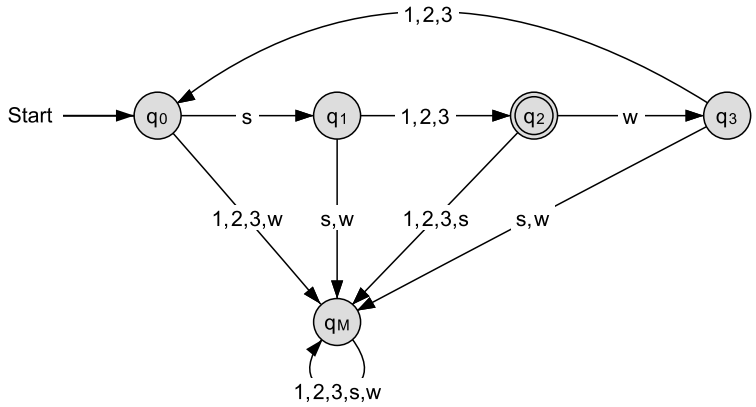
	`int anzahl = aufnahmestapel.size();`
	`for (int i = 0; i < anzahl; i++) {`
	`  indexZufall = (int) (anzahl * Math.random());`
	`  karte = aufnahmestapel.get(i);`
	`  karteZufall = aufnahmestapel.get(indexZufall);`
	`  aufnahmestapel.set(i, karteZufall);`
	`  aufnahmestapel.set(indexZufall, karte);`
	`}`			

2.2.4	`Karte k = handkarten.get(pIndex)`
	`ablagestapel.add(k);`
	`handkarten.remove(pIndex);`
	Hinweis: Eine Bedingungsprüfung für `pIndex` ist nicht notwendig.
		0	1	2

2.3	**Kartenprüfgeräte**			
2.3.1	elektronisch arbeitend, Nutzung des Binärsystems, getaktete Arbeitsweise, speicherprogrammiert, sequenzielle Abarbeitung von Befehlen, von-Neumann-Zyklus, …	2	0	0
2.3.2	Speicherzelle 0 enthält den Befehl `NULL 010`, das bedeutet, dass die Zahl 0 in die Speicherzelle 10 geschrieben wird. Speicherzelle 1 enthält den Befehl `TST 011`. Hierdurch wird der Inhalt der Zelle 11 (hier die Zahl 7) mit der Zahl 0 verglichen. Bei Gleichheit wird der folgende Befehl übersprungen (hier `JMP 004`) und der übernächste Befehl ausgeführt (hier `HLT`). Da 7 ungleich 0 ist, wird der Befehl `JMP 004` ausgeführt.	1	2	0
2.3.3	Das Programm zählt die Anzahl der ermittelten Qualitätswerte und damit die Anzahl der gescannten Karten. Ermittlung beispielsweise mithilfe einer Speicherbelegungstabelle wie folgt: Zelle 01: 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 Zelle 10: 10	0	0	2
	gesamt	**9**	**15**	**6**

3 Wahlaufgabe: Strichcodes

<table>
<tr><th>3.1</th><th>Strichcode und Automat</th><th>I</th><th>II</th><th>III</th></tr>
<tr><td>3.1.1</td><td>s1w2s1w1s3w2s2
 

Hinweis: Die Angabe der Strichbreite unter der Grafik ist nicht notwendig.</td><td>2</td><td>0</td><td>0</td></tr>
<tr><td>3.1.2</td><td>s2w1w2s3: Wechsel der Strichfarbe nicht beachtet
s2w2: Code endet nicht mit einem schwarzen Strich</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>3.1.3</td><td>s2w1w2s3: $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_1 \rightarrow q_2 \rightarrow q_1 \rightarrow q_2 \rightarrow q_1 \rightarrow q_2$ ➔ Endzustand
s2w2: $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_1 \rightarrow q_2$ ➔ Endzustand</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td>3.1.4</td><td> 
</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td>3.2</td><td>Softwareprojekt</td><td></td><td></td><td></td></tr>
<tr><td>3.2.1</td><td>Der Konstruktor erzeugt Objekte und weist den Attributen initiale Werte zu.
Dem Attribut <code>strichcode</code> wird <code>s1</code> zugewiesen. Den Attributen <code>erweiterterStrichcode</code> und <code>alternativerStrichcode</code> wird eine leere Zeichenkette zugewiesen.</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>3.2.2</td><td><code>erweiterterStrichcode = strichcode + "w4"</code> → s2w2s1w2s3w4
<code>strichcode.length() = 10</code>
<table>
<tr><td>i</td><td>strichangabe</td><td>erweiterterStrichcode</td></tr>
<tr><td></td><td></td><td>s2w2s1w2s3w4</td></tr>
<tr><td>8</td><td>s3</td><td>s2w2s1w2s3w4s3</td></tr>
<tr><td>6</td><td>w2</td><td>s2w2s1w2s3w4s3w2</td></tr>
<tr><td>4</td><td>s1</td><td>s2w2s1w2s3w4s3w2s1</td></tr>
<tr><td>2</td><td>w2</td><td>s2w2s1w2s3w4s3w2s1w2</td></tr>
<tr><td>0</td><td>s2</td><td>s2w2s1w2s3w4s3w2s1w2s2</td></tr>
<tr><td>-2</td><td></td><td></td></tr>
</table></td><td>0</td><td>2</td><td>2</td></tr>
</table>

	Der gegebene Strichcode wird um `w4` erweitert; anschließend werden die Blöcke von Farbe und Strichbreite in umgekehrter Reihenfolge angefügt.			

3.2.3	1. Umsetzung der Zählschleife
	2. Bestimmung von `farbe` und `anzahl`
	3. Zweiseitige Verzweigungen
	4. Erzeugung von `code` und Festlegung von `alternativerStrichcode`
	`for (int i = 0; i < strichcode.length(); i = i+2) {`
	`  farbe = strichcode.charAt(i);`
	`  anzahl = strichcode.charAt(i+1);`
	`  if (anzahl == '1') {code = code + farbe;}`
	`  else {`
	`  if (anzahl == '2') {code = code + farbe + farbe;}`
	`  else {code = code + farbe + farbe + farbe;}`
	`  }`
	`}`
	`alternativerStrichcode = code;`
		0	3	1

3.2.4	Alternative A, z. B.: • Darstellung länger als Darstellung nach ursprünglichem Codierprinzip (Nachteil) • Jedes Zeichen kann direkt in einen schwarzen oder weißen Teilstrich abgebildet werden (Vorteil) Alternative B, z. B.: • Darstellung kürzer als Darstellung nach ursprünglichem Codierprinzip (Vorteil) • Zusätzliches Wissen erforderlich (z. B. Strichcode beginnt mit einem schwarzen Strich)	0	2	2
3.3	**Datenübertragung**			
3.3.1	Router: Kopplung von Netzen sowie Weiterleitung von Netzwerkpaketen anhand ihrer IP-Adressen zu anderen Netzen oder Endgeräten Router arbeiten auf der Vermittlungsschicht (OSI-Modell) bzw. der Internetschicht (DoD-TCP/IP-Modell) Server: Anbieten von Diensten, Steuerung und Organisation der Kommunikation, Speicherung und Verwaltung von Programmen und Daten	3	0	0
3.3.2	Aus dem Suffix 16 folgt die Netzmaske 255.255.0.0. Die Netzmaske maskiert über die gesetzten ersten 16 Bits Netz- und Hostteil. Dadurch entsteht die Netzadresse 87.142.0.0 und der Hostteil 80.212 = 20692.	2	1	0
	gesamt	**9**	**15**	**6**