

Abiturprüfung 2013

INFORMATIK

Hinweise zur Korrektur und Bewertung

(nicht für den Prüfling bestimmt)

Die Korrekturhinweise enthalten keine vollständigen Lösungen der Aufgaben, sondern einen Abriss des Erwartungshorizontes. Nicht genannte, aber *gleichwertige* Lösungswege und Begründungsansätze sind *gleichberechtigt*.

Die in den Lösungshinweisen gewählte objektorientierte Programmiersprache ist Java.

Die Bewertung der erbrachten Prüfungsleistungen hat sich an der bei jeder Teilaufgabe am linken Rand des Angabenblattes vermerkten, maximal erreichbaren Zahl von Bewertungseinheiten (BE) zu orientieren.

Umrechnung der erreichten Bewertungseinheiten in Notenpunkte:

Die insgesamt erreichten Bewertungseinheiten werden nach der folgenden Tabelle in Notenpunkte umgesetzt:

Notenpunkte	Notenstufen	Bewertungseinheiten	Intervalle in %
15	+1	120 ... 115	15
14	1	114 ... 109	
13	1–	108 ... 103	
12	+2	102 ... 97	15
11	2	96 ... 91	
10	2–	90 ... 85	
9	+3	84 ... 79	15
8	3	78 ... 73	
7	3–	72 ... 67	
6	+4	66 ... 61	15
5	4	60 ... 55	
4	4–	54 ... 49	
3	+5	48 ... 41	20
2	5	40 ... 33	
1	5–	32 ... 25	
0	6	24 ... 0	20

Aufgabe	BE	Hinweise
1. a)	12	Attributdeklarationen: <pre> private String name; private int kundennummer; private String adresse; private Dvd[] wuensche; private int anzahlWuensche; private Dvd entlieheneDVD; </pre> geforderte Methode: <pre> void wunschdvdAnfordern(){ int i = 0; if (entlieheneDVD != null){ System.out.println("bereits eine DVD entliehen"); }else{ while ((i < anzahlWuensche) && (wuensche[i].istEntliehen())){ i++; } if (i == anzahlWuensche){ System.out.println("keine entleihbare DVD gefunden"); }else{ wuensche[i].entleiherEintragen(this); entlieheneDVD = wuensche[i]; wunschLoeschen(i); } } } </pre>
b)	10	<pre> sequenceDiagram participant k as k:KUNDE participant d1 as d1:DVD participant d2 as d2:DVD k->>k: wunschdvdAnfordern() activate k k->>d1: istEntliehen() activate d1 d1-->>k: true deactivate d1 k->>d2: istEntliehen() activate d2 d2-->>k: false deactivate d2 k->>d2: entleiherEintragen(k) activate d2 d2-->>k: deactivate d2 k->>k: wunschLoeschen(1) activate k k-->>k: deactivate k </pre> <i>Hinweis:</i> Je nach Implementierung der Zählung in Teilaufgabe 1a muss statt <i>wunschLoeschen(1)</i> ggf. <i>wunschLoeschen(2)</i> aufgerufen werden.

		Ein Problem ergibt sich zum Beispiel, wenn ein weiterer Kunde nach dem obigen zweiten Aufruf von <i>istEntliehen</i> bei dieser DVD <i>entleiherEintragen</i> aufruft. Vermeiden ließe sich dieses Problem, wenn z. B. durch ein Semaphore sichergestellt würde, dass der Zugriff eines Kundenobjekts auf eine DVD nicht unterbrochen werden darf.
--	--	--

2. a)	6	<pre> classDiagram class DVDLISTE class LISTENELEMENT { <abstract> } class ABSCHLUSS class DATENKNOTEN class DVD DVDLISTE "1" -- "1" LISTENELEMENT : erstes > 1 LISTENELEMENT "1" -- "1" DATENKNOTEN LISTENELEMENT "1" -- "1" LISTENELEMENT : naechstes ABSCHLUSS -- > LISTENELEMENT DATENKNOTEN -- > LISTENELEMENT DATENKNOTEN "0..n" -- "1" DVD : inhalt </pre>
-------	---	--

b)	14	<pre> class DvdListe{ private Listenelement erstes; DvdListe(){ erstes = new Abschluss(); } void dvdHinzufuegen(Dvd d){ erstes = new Datenknoten(d, erstes); } void alleTitelAnzeigen(){ erstes.alleTitelAnzeigen(); } } abstract class Listenelement{ abstract void alleTitelAnzeigen(); } </pre>
----	----	--

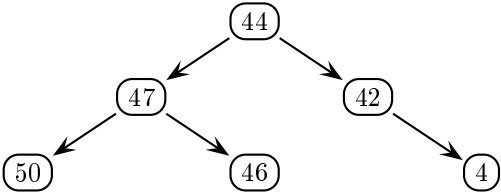
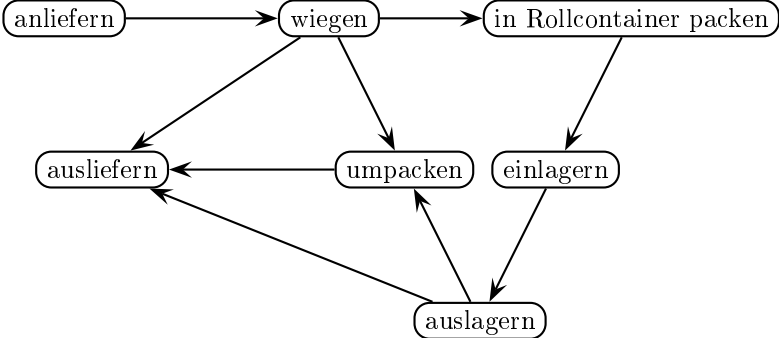
		<pre> class Datenknoten extends Listenelement{ private Listenelement naechstes; private Dvd inhalt; Datenknoten(Dvd eineDVD, Listenelement n){ inhalt = eineDVD; naechstes = n; } void alleTitelAnzeigen(){ naechstes.alleTitelAnzeigen(); System.out.println(inhalt.titelGeben()); } } class Abschluss extends Listenelement{ void alleTitelAnzeigen(){ } } </pre>
3. a)	5	<pre> graph TD Batman --> MonacoFranze[Monaco Franze] MonacoFranze --> HighNoon[High Noon] MonacoFranze --> Titanic[Titanic] HighNoon --> FluchDerKaribik[Fluch der Karibik] HighNoon --> KonigDerLoewen[König der Löwen] FluchDerKaribik --> Cars3[Cars 3] </pre> Ausbalanciert wäre der Baum z. B. bei der Reihenfolge High Noon, Cars 3, Monaco Franze, Fluch der Karibik, König der Löwen, Batman, Titanic.
b)	4	Inorder-Traversierung: Alle im Baum gespeicherten DVD-Titel in alphabetischer Reihenfolge können durch eine rekursive Methode, die in jedem Knoten jeweils erst den linken Teilbaum, dann den Inhalt und schließlich den rechten Teilbaum bearbeitet, ausgegeben werden.
c)	7	Der Baum sollte möglichst geringe Höhe haben. Hier sind das 15 Ebenen. Begründung: in einem Baum mit 1 Ebene kann $1 = 2^1 - 1$ Element gespeichert werden, in einem Baum mit 2 Ebenen können $1 + 2 = 3 = 2^2 - 1$ Elemente gespeichert werden, in einem Baum mit 3 Ebenen können $1 + 2 + 4 = 7 = 2^3 - 1$ Elemente gespeichert werden, ... in einem Baum mit k Ebenen können $2^k - 1$ Elemente gespeichert werden. $2^{14} - 1 = 16383$, $2^{15} - 1 = 32767$, also sind für 20000 Elemente 15 Ebenen nötig. Eine Suche dauert also maximal $1 \text{ ns} \cdot 15 = 15 \text{ ns}$.

4. a)	4	<table><tr><td></td><td>dvd1</td><td>dvd2</td><td>dvd3</td><td>dvd4</td><td>dvd5</td></tr><tr><td>dvd1</td><td></td><td>2</td><td>9</td><td></td><td></td></tr><tr><td>dvd2</td><td>2</td><td></td><td></td><td>27</td><td></td></tr><tr><td>dvd3</td><td>9</td><td></td><td></td><td>3</td><td></td></tr><tr><td>dvd4</td><td></td><td>27</td><td>3</td><td></td><td>1</td></tr><tr><td>dvd5</td><td></td><td></td><td></td><td>1</td><td></td></tr></table>		dvd1	dvd2	dvd3	dvd4	dvd5	dvd1		2	9			dvd2	2			27		dvd3	9			3		dvd4		27	3		1	dvd5				1	
	dvd1	dvd2	dvd3	dvd4	dvd5																																	
dvd1		2	9																																			
dvd2	2			27																																		
dvd3	9			3																																		
dvd4		27	3		1																																	
dvd5				1																																		
b)	3	<pre>graph LR dvd1 --- 2 dvd2 dvd1 --- 9 dvd3 dvd2 --- 1 dvd3 dvd2 --- 28 dvd4 dvd3 --- 3 dvd4 dvd4 --- 1 dvd5</pre>																																				
c)	4	Ein Algorithmus zum Graphdurchlauf ist die Tiefensuche. Nicht alle Knoten werden besucht, wenn z. B. eine DVD nie entliehen worden ist. Alternativ kann auch argumentiert werden, dass sich zwei Teilgraphen z. B. aus Kinder- und Action-DVDs gebildet haben, solange die Kunden immer nur entweder Kinder- oder Action-DVDs entliehen haben.																																				
d)	11	$m3$ ermittelt zunächst – durch Aufruf von $m1$ – das Maximum u aller Kantengewichte der Kanten, die vom Knoten mit dem Index k ausgehen. Ist dieses nicht 0, dann werden – durch Aufruf von $m2$ – die Titel aller DVDs angezeigt, die mit der DVD mit dem Index k durch eine Kante mit Kantengewicht u verbunden sind. Es werden alle DVDs angezeigt, die am häufigsten mit der angegebenen DVD in einer Historie standen.																																				
	80																																					

Aufgabe	BE	Hinweise
1. a)	3	Das Pflichtenheft ist eine aus technischer Sicht detaillierte, eindeutige und testbare Beschreibung der vom Auftraggeber (im Lastenheft) vorgegebenen Systemanforderungen. Es wird in der Analysephase erarbeitet.
b)	2	Beim Entwurf von objektorientierten Programmen (oder im größeren Rahmen von Softwaresystemen) versucht man, auf bereits bewährte Lösungen ähnlicher Aufgabenstellungen zurückzugreifen. Die Beschreibung solcher wieder verwendbaren Lösungsansätze, beispielsweise in Form eines Klassendiagramms, nennt man Softwaremuster.
c)	7	Model-View-Controller (MVC) ist ein Softwaremuster zur Strukturierung von Softwaresystemen in die drei Einheiten (Daten-)Modell (engl. model), Präsentation (engl. view) und Programmsteuerung (engl. controller). • Das Modell enthält die zu verwaltenden Daten. • Die Präsentationsschicht ist für die Darstellung der benötigten Daten aus dem Modell und die Entgegennahme von Benutzerinteraktionen zuständig. • Die Steuerung verwaltet eine oder mehrere Präsentationen, nimmt von ihnen Benutzeraktionen entgegen, wertet diese aus und agiert entsprechend. Ein konkreter Vorteil ist der relativ einfach zu bewerkstellende Austausch von Komponenten. Beispielsweise kann die GUI durch ein Webinterface ersetzt werden, ohne in Logik und Datenmodell eingreifen zu müssen.
2. a)	4	Schlange, Warteschlange, Queue Diese Datenstruktur hat zwei zentrale Methoden, nämlich <i>einfüegen(datenobjekt)</i> und <i>entfernen()</i>. Erstere fügt ein Datenobjekt an dem einen Ende der Schlange ein, während letztere das Element am anderen Ende der Schlange entfernt.
b)	3	Mögliches Beispiel: Das Hochregal ist leer; Container A wird in Fach 1 eingelagert, danach Container B in Fach 2; Container B wird dann vor Container A entnommen, was möglich ist, weil sich die Container in unterschiedlichen Fächern befinden. Damit wird also der Container, der als zweiter eingelagert wurde, als erster wieder aus dem Hochregal entnommen.
c)	2	Mögliches Beispiel: Regal, in dem die Container hintereinander stehen und das nur von einer Seite beladen und wieder entladen werden kann.

3.	12	<pre> class Durchlaufregal{ private Fach[] faecher; Durchlaufregal(int fachanzahl){ faecher = new Fach[fachanzahl]; for (int i = 0; i<fachanzahl; i++){ faecher[i]= new Fach(i+1); //Übergeben wird bei dem hier verwendeten Konstruktor die Fachnummer. } } boolean einlagernMoeglich(double containerlaenge, int fachnr){ return (faecher[fachnr-1].nochFreieLaengeGeben() >= containerlaenge); } void einlagern(Container container, int fachnr){ faecher[fachnr-1].einfuegen(container); //Die Nummerierung der Fächer beginnt bei 1, der Feldindex bei 0. } } </pre>
4.	4	Mögliche Aspekte: • Containeranzahl Da die Länge der einzulagernden Container nicht bekannt ist, kann einerseits nicht genau vorhergesagt werden, wie viele Container in einem Fach (maximal) abgestellt werden können. Andererseits können nicht beliebig viele Container in ein Fach eingefügt werden, so dass man sicher ein genügend großes Feld instanziiieren könnte, ohne zu viel Speicherplatz zu verschwenden. • Entnahme eines Containers aus dem Fach Da eine Warteschlange umgesetzt wird, muss einerseits beim Entfernen des ersten Elements das Feld reorganisiert werden, was mit Aufwand verbunden ist. Andererseits ist die Implementierung mit einem Feld relativ einfach. Je nach Gewichtung der Argumente kann sich der Schüler für oder gegen eine Eignung aussprechen.
5. a)	7	<pre> classDiagram class LISTE class LISTENELEMENT { <<abstract>> } class ABSCHLUSS class KNOTEN class DATENELEMENT { <<abstract>> } class CONTAINER class FACH LISTE "1" -- "1" LISTENELEMENT : erstesElement > LISTENELEMENT < -- ABSCHLUSS LISTENELEMENT < -- KNOTEN KNOTEN "1" -- "1" LISTENELEMENT : nachfolger ^ KNOTEN "1" -- "1" DATENELEMENT : verwaltet v DATENELEMENT < -- CONTAINER LISTE "1" -- "1" FACH : nutzt ^ </pre>

b)	11	<pre> class Liste{ private Listenelement erstesElement; int anzahlGeben(){ return erstesElement.anzahlAbHierGeben(); } } abstract class Listenelement{ abstract int anzahlAbHierGeben(); } class Knoten extends Listenelement{ private Listenelement nachfolger; int anzahlAbHierGeben(){ return 1 + nachfolger.anzahlAbHierGeben(); } } class Abschluss extends Listenelement{ int anzahlAbHierGeben() { return 0; } } </pre>
c)	6	<pre> sequenceDiagram participant l as l:LISTE participant k1 as k1:KNOTEN participant k2 as k2:KNOTEN participant k3 as k3:ABSCHLUSS l->>k1: anzahlAbHierGeben() activate k1 k1->>k2: anzahlAbHierGeben() activate k2 k2->>k3: anzahlAbHierGeben() activate k3 k3-->>k2: 0 deactivate k3 k2-->>k1: 1(=1+0) deactivate k2 k1-->>l: 2(=1+1) deactivate k1 </pre>
6. a)	2	Für jeden Knoten des Baums gilt: Im linken Teilbaum werden die Knoten mit größerem Schlüssel als dem Knotenschlüssel gespeichert, im rechten Teilbaum die mit kleinerem Schlüssel.

b)	5	 Ausgehend von der Wurzel muss die ID 50 im linken Teilbaum, die ID 4 aber im rechten Teilbaum einsortiert werden. Damit wird auf voneinander unabhängigen Teilbäumen gearbeitet.
c)	7	Die Klasse KNOTEN enthält drei Referenzattribute: • linkerNachfolger mit Referenz auf den linken Nachfolgerknoten • rechterNachfolger mit Referenz auf den rechten Nachfolgerknoten • daten mit Referenz auf das Container-Objekt Methode spezialdatenAusgeben() rechterNachfolger.spezialdatenAusgeben() wenn daten.containerlaengeGeben() $\geq$ 1 Meter dann daten.datenAufKonsoleAusgeben() endeWenn linkerNachfolger.spezialdatenAusgeben() endeMethode
7.	5	
	80	

Aufgabe	BE	Hinweise
1. a)	2	§012 wird nicht erkannt §1109(2) wird erkannt (Endzustand z7) §8 wird erkannt (Endzustand z3) §31(180) wird nicht erkannt
b)	6	EBNF: paragraph = '§' zifferNichtNull { ziffer } [klammerausdruck]; klammerausdruck = '(' zifferNichtNull [ziffer] ')'; ziffer = '0' zifferNichtNull; zifferNichtNull = '1' '2' ... '9'; Syntaxdiagramm:
2. a)	8	<i>Hinweis:</i> Der Fehlerzustand kann weggelassen werden.
b)	3	Z. B.: Ein anderer Sachbearbeiter mit dem Kürzel MSP könnte den Vorgang abschließen. Der betreffende Dateiname wird fälschlicherweise von obigem Automaten nicht akzeptiert.

3. a)	3	fib(5) = fib(4) + fib(3) = fib(3) + fib(2) + fib(2) + fib(1) = fib(2) + fib(1) + 1 + 1 + 1 = 1 + 1 + 1 + 1 + 1 = 5																																								
b)	7	fib wurde in Teilaufgabe 3a insgesamt 9 mal aufgerufen fib(3) = fib(2) + fib(1) = 1 + 1 a(3) = 1+a(2)+a(1)=3 Aufrufe fib(4) = fib(3) + fib(2) a(4) = 1+a(3)+a(2)=1+3+1 = 5 Aufrufe fib(5) = fib(4) + fib(3) a(5) = 1+a(4)+a(3)=1+5+3 = 9 Aufrufe ... <table><tr><th>n</th><th>a(n)</th><th>a(n) / a(n-1)</th></tr><tr><td>1</td><td>1</td><td>--</td></tr><tr><td>2</td><td>1</td><td>1</td></tr><tr><td>3</td><td>3</td><td>3</td></tr><tr><td>4</td><td>5</td><td>1,67</td></tr><tr><td>5</td><td>9</td><td>1,8</td></tr><tr><td>6</td><td>15</td><td>1,67</td></tr><tr><td>7</td><td>25</td><td>1,67</td></tr><tr><td>8</td><td>41</td><td>1,64</td></tr></table> Da der Quotient a(n) / a(n-1) etwa bei 1,6 liegt, die Anzahl der Aufrufe von fib sich mit jeder Erhöhung von n um 1 also etwa ver-1,6-facht, liegt vermutlich ein exponentielles Laufzeitverhalten mit Basis von ungefähr 1,6 vor.	n	a(n)	a(n) / a(n-1)	1	1	--	2	1	1	3	3	3	4	5	1,67	5	9	1,8	6	15	1,67	7	25	1,67	8	41	1,64													
n	a(n)	a(n) / a(n-1)																																								
1	1	--																																								
2	1	1																																								
3	3	3																																								
4	5	1,67																																								
5	9	1,8																																								
6	15	1,67																																								
7	25	1,67																																								
8	41	1,64																																								
4. a)	5	<table><tr><th>Befehl</th><th>Akk</th><th>101</th><th>102</th></tr><tr><td></td><td></td><td>5</td><td>18</td></tr><tr><td>load 102</td><td>18</td><td></td><td></td></tr><tr><td>div 101</td><td>3</td><td></td><td></td></tr><tr><td>mult 101</td><td>15</td><td></td><td></td></tr><tr><td>sub 102</td><td>-3</td><td></td><td></td></tr><tr><td>jeq 8</td><td></td><td></td><td></td></tr><tr><td>dload 0</td><td>0</td><td></td><td></td></tr><tr><td>jump 9</td><td></td><td></td><td></td></tr><tr><td>end</td><td></td><td></td><td></td></tr></table> Das Programm überprüft, ob der Wert in 102 ein Vielfaches des Wertes in 101 ist (dann steht 1 im Akkumulator) oder nicht (dann steht 0 im Akkumulator).	Befehl	Akk	101	102			5	18	load 102	18			div 101	3			mult 101	15			sub 102	-3			jeq 8				dload 0	0			jump 9				end			
Befehl	Akk	101	102																																							
		5	18																																							
load 102	18																																									
div 101	3																																									
mult 101	15																																									
sub 102	-3																																									
jeq 8																																										
dload 0	0																																									
jump 9																																										
end																																										

b)	6	<pre> 1: dload 5 -- wähle z.B. für n die Zahl 5 2: store 202 3: dload 1 4: store 201 5: load 202 6: jle 13 7: mult 201 8: store 201 9: load 202 10: dec 11: store 202 12: jump 5 13: end </pre> <i>Hinweis:</i> Die Zeilen 1 und 2 können entfallen und durch die Angabe der Vorbelegung der Zellen, die vor dem Programmstart gegeben ist, ersetzt werden.
	40	

Aufgabe	BE	Hinweise
1.	11	
2. a)	6	Bei der Eingabe der Zeichenkette AUGUGGUGGUGA werden der Reihe nach die Zustände 0, 3, 4, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0 eingenommen. Der folgende Text wird ausgegeben: "Das Wort wurde erkannt: 2"
b)	5	Es werden alle Zeichenketten erkannt, die aus Tripeln der Buchstaben A, C, G oder U bestehen. Im Attribut x wird gezählt, wie oft unter diesen das Tripel UGG vorkommt. <i>Hinweis:</i> Als Teillösung darf auch der entsprechende Automat (vgl. Abbildung) in die Bewertung eingehen.

3.	7	<pre> ... 07: load 101 08: store 102 -- y in Zelle 102 09: dload 1 10: store 103 -- z in Zelle 103 11: load 103 -- z holen 12: sub 102 -- y subtrahieren 13: jge 22 -- falls y <= z 14: load 102 -- y holen 15: add 103 -- z dazu addieren 16: div 104 -- durch 2 dividieren 17: store 102 -- y = (y+z)/2 18: load 101 -- V holen 19: div 102 -- durch y dividieren 20: store 103 -- z = V / y 21: jump 11 -- nächste Wiederholung starten 22: load 102 -- y holen 23: div 105 -- durch 19 dividieren 24: store 106 -- Rückgabe in Zelle 106 25: end </pre> <i>Hinweis:</i> Die Kommentare sind nicht verlangt.
4. a)	4	Synchronisation bedeutet, dass nebenläufige Prozesse, welche auf gemeinsame Ressourcen zugreifen, zeitlich abgestimmt werden, wie z. B. die Herstellung der beiden Packungsgrößen X und Y durch die drei Roboter (gemeinsame Ressource). Ein kritischer Abschnitt ist der Teilabschnitt der Prozessbeschreibung, in dem ein Prozess auf eine mit einem anderen Prozess gemeinsame Ressource zugreift. Im Beispiel ist jeder Arbeitsschritt kritisch, z. B. der Arbeitsschritt X2.

b)

7

Zwei oder mehrere Prozesse befinden sich in Verklemmung, wenn jeder dieser Prozesse auf ein Ereignis wartet, das nur von einem anderen dieser Prozesse ausgelöst werden kann. Im Beispiel liegt eine Verklemmung vor, wenn Prozess X den Roboter R1 belegt hat und für den zweiten Schritt auf R2 wartet. Gleichzeitig hat Prozess Y den Roboter R2 belegt und wartet im zweiten Arbeitsschritt auf R1, um den Vorgang abschließen zu können.

