

I. Erläuterungen

Aufgabenart

materialgebundene Aufgabenstellung

Voraussetzungen gemäß Lehrplan und Erlass „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2014“ vom 20. Juni 2012

Q1 objektorientierte Modellierung

Objektmodell, Klassen, iterative Verfahren, Grundkonzepte des Software-Engineerings

II. Lösungshinweise und Bewertungsraster

In den nachfolgenden Lösungshinweisen sind alle wesentlichen Gesichtspunkte, die bei der Bearbeitung der einzelnen Aufgaben zu berücksichtigen sind, konkret genannt und diejenigen Lösungswege aufgezeigt, welche die Prüflinge erfahrungsgemäß einschlagen werden. Selbstverständlich sind jedoch Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, ebenso zu akzeptieren.

Aufg.	erwartete Leistungen	BE			
		I	II	III	Σ
1.1	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Philosoph</p> <hr/> <ul style="list-style-type: none"> <input type="checkbox"/> Name: String <input type="checkbox"/> denkend: boolean <input type="checkbox"/> hungrig: boolean <input type="checkbox"/> essend: boolean <input type="checkbox"/> linkeGabel: Gabel <input type="checkbox"/> rechteGabel: Gabel <input type="checkbox"/> hatGabeln: int <hr/> <ul style="list-style-type: none"> ⊙ Philosoph(Name: String, linkeGabel: Gabel, rechteGabel: Gabel) ⊕ getName(): String ⊕ getLinkeGabel(): Gabel ⊕ denken() ⊕ hungern() ⊕ essen() ⊕ nimmLinkeGabel() ⊕ nimmRechteGabel() </div> <p style="text-align: center;">↓</p> <div style="border: 1px solid black; padding: 5px; margin-left: auto; margin-right: auto;"> <p style="text-align: center;">Gabel</p> <hr/> <ul style="list-style-type: none"> <input type="checkbox"/> aufDemTisch: boolean <hr/> <ul style="list-style-type: none"> ⊙ Gabel() ⊕ getAufDemTisch(): boolean ⊕ setAufDemTisch(aufDemTisch: boolean) </div>	7	5		12
1.2	Bei der Beziehung zwischen den beiden Klassen handelt es sich um eine Assoziation bzw. kennt-Beziehung. Objekte der Klasse <i>Philosoph</i> kennen je zwei Objekte der Klasse <i>Gabel</i> . Dies ist in den Attributen <i>linkeGabel</i> und <i>rechteGabel</i> vom Datentyp <i>Gabel</i> realisiert.	3	2		5

Aufg.	erwartete Leistungen	BE			
		I	II	III	Σ
2	<pre>public void nimmLinkeGabel() { System.out.print(Name + " versucht seine linke Gabel zu nehmen: "); if (linkeGabel.getAufDemTisch()) { linkeGabel.setAufDemTisch(false); System.out.println("erfolgreich."); hatGabeln++; } else { System.out.println("wartet."); } }</pre>	4	3		7
3	<pre>Gabel G0 = new Gabel(); Gabel G1 = new Gabel(); Gabel G2 = new Gabel(); Philosoph Sokrates = new Philosoph("Sokrates", G1, G0); Philosoph Platon = new Philosoph("Platon", G2, G1); Sokrates.hungern(); Sokrates.nimmLinkeGabel(); Sokrates.nimmRechteGabel(); Sokrates.essen(); Platon.hungern(); Platon.nimmLinkeGabel(); Platon.nimmRechteGabel(); Sokrates.denken(); Platon.nimmRechteGabel(); Platon.essen();</pre>	4	3		7
4.1	Das Problem besteht darin, dass nun jeder Philosoph seine linke Gabel hält und somit kein Philosoph seine rechte Gabel aufnehmen kann, um zu essen. Da die Folge der Aktionen festgelegt ist, kann kein Philosoph diesen Zustand verlassen und es kommt zum Stillstand.		3	2	5
4.2	<p>Die Methode <i>geprüftLinkeGabelNehmen()</i> wird mit dem Namen eines Philosophen als Parameter aufgerufen. Für diesen Philosophen wird geprüft, ob sein linker Platznachbar bereits seine linke Gabel hält. Nur wenn dies nicht der Fall ist, nimmt er seine linke Gabel auf.</p> <p>Dafür wird zunächst der Philosoph im Feld gesucht. Dies erfolgt in einer while-Schleife, die solange die Position <i>i</i> erhöht, wie das Feld noch Plätze hat und der gesuchte Philosoph noch nicht gefunden ist. Die Schleife endet, wenn <i>i</i> gleich der Anzahl der Plätze ist oder der gesuchte Philosoph am Platz <i>i</i> sitzt.</p> <p>Sollte <i>i</i> gleich der Anzahl der Plätze sein, erfolgt eine entsprechende Fehlermeldung. Andernfalls wird in <i>linkerNachbar</i> die Position des linken Nachbarn gespeichert. Ist diese Position gleich <i>AnzahlPlätze</i> wird <i>linkerNachbar</i> auf 0 gesetzt.</p> <p>Anschließend wird für den linken Sitznachbarn abgefragt, ob dessen linke Gabel auf dem Tisch liegt. Wenn das der Fall ist, kann der Philosoph seine linke Gabel aufnehmen, andernfalls erfolgt eine entsprechende Meldung.</p>		7	5	12

Aufg.	erwartete Leistungen	BE			
		I	II	III	Σ
4.3	Versucht ein Philosoph erfolglos seine rechte Gabel aufzunehmen, müsste es die Möglichkeit geben, dass er die linke Gabel wieder zurücklegt ohne gegessen zu haben. Oder: Die Philosophen benötigen eine Assoziation zu mindestens einem Sitznachbarn: Bevor ein Philosoph seine linke Gabel ergreift, fragt er bei seinem linken Nachbarn an, ob dieser schon seine linke Gabel hält. Sollte dies der Fall sein, nimmt er nicht seine linke Gabel auf, sondern wartet. Alternativ könnte auch jeder Philosoph seinen rechten Nachbarn anfragen und sich bei ihm, wenn gerade möglich, seine rechte Gabel reservieren.			5	5
5	<pre> public void zeigeFeld() { for (int i = 0; i < AnzahlPlätze; i++) { if (Philosophen[i] != null) { System.out.println("An Platz " + i + " sitzt Philosoph " + Philosophen[i].getName() + "."); } else { System.out.println("Platz " + i + " ist frei."); } } } </pre>		7		7
		18	30	12	60

III. Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt gemäß den Bestimmungen in der OAVO in der jeweils gültigen Fassung, insbesondere § 33 OAVO in Verbindung mit den Anlagen 9a und ggf. 9b bis 9f, sowie in den Einheitlichen Prüfungsanforderungen in der Abiturprüfung (EPA). Für die Umrechnung von Prozentanteilen der erbrachten Leistungen in Notenpunkte nach § 9 Abs. 12 der OAVO gelten die Werte in der Anlage 9a der OAVO. Darüber hinaus sind die Vorgaben des Erlasses „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2014“ vom 20. Juni 2012 zu beachten.

Bei der Bewertung und Beurteilung ist auch die Intensität der Bearbeitung zu berücksichtigen. Als Bewertungskriterien dienen über das Inhaltliche hinaus qualitative Merkmale wie Strukturierung, Differenziertheit und Schlüssigkeit der Argumentation.

Im Fach Informatik (Grundkurs) können Vorschläge zu den Themen der drei Kurshalbjahre Q1 (objekt-orientierte Modellierung), Q2 (Datenbanken) und Q3 (Konzepte und Anwendungen der theoretischen Informatik) vorgelegt werden. Die Prüfungsleistung besteht aus der Bearbeitung von zwei Vorschlägen, einem zum Halbjahresthema „objektorientierte Modellierung“ und einem weiteren zu einem der beiden anderen Halbjahresthemen, wofür insgesamt maximal 100 BE vergeben werden können. Ein Prüfungsergebnis von **5 Punkten** (ausreichend) setzt voraus, dass insgesamt 46 BE, ein Prüfungsergebnis von **11 Punkten** (gut), dass insgesamt 76 BE erreicht werden.

Gewichtung der Aufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen

Aufgabe	Bewertungseinheiten in den Anforderungsbereichen			Summe
	AFB I	AFB II	AFB III	
1.1	7	5		12
1.2	3	2		5
2	4	3		7
3	4	3		7
4.1		3	2	5
4.2		7	5	12
4.3			5	5
5		7		7
Summe	18	30	12	60

Die auf die Anforderungsbereiche verteilten Bewertungseinheiten innerhalb der Aufgaben sind als Richtwerte zu verstehen.