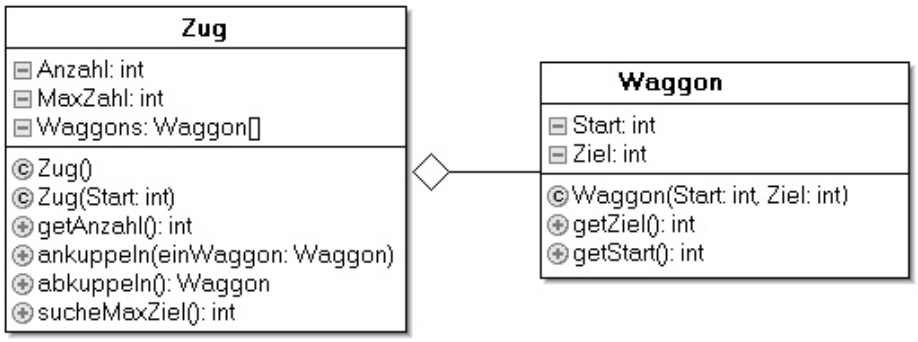
I. Erläuterungen

Aufgabenart
Objektorientierte Modellierung

Voraussetzungen gemäß Lehrplan und Erlass „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2013“ vom 20. Juni 2011
Objektmodell, Klassen
Sortieralgorithmen, iterative Verfahren

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
1.1	 Aggregation, da gemäß Aufgabe ein Zug aus keinem, einem oder mehreren Waggons besteht.	5	7	
1.2	*public Zug()*: Das Feld *Waggons* wird mit einer for-Schleife durchlaufen, wobei jedes Feldelement mit *null* initialisiert wird. Abschließend wird die Anzahl der Waggons auf 0 gesetzt. *public Zug(int Start)*: Hier wird für jedes Feldelement der Konstruktor der Klasse *Waggon* mit dem Parameter *Start* sowie verschiedenen Werten für *Ziel* aufgerufen. Abschließend wird *Anzahl* mit *MaxZahl* belegt.	3	3	

2.1	`public void ankuppeln(Waggon einWaggon) {`
	`  if (Anzahl < MaxZahl) {`
	`  for (int i = Anzahl; i > 0; i--)`
	`  Waggons[i] = Waggons[i-1];`
	`  Waggons[0] = einWaggon;`
	`  Anzahl++;`
	`  }`
	`}`
		2	4	2

Aufg.	erwartete Leistungen	BE		
		I	II	III

2.2	`public int sucheMaxZiel() {`
	`  int MaxZiel = -1;`
	`  for (int i = 0; i < getAnzahl(); i++)`
	`  if (MaxZiel < Waggons[i].getZiel())`
	`  MaxZiel = Waggons[i].getZiel();`
	`  return MaxZiel;`
	`}`
			4	4

3.1	*WaggonsAbstossen()*: Zunächst wird mit *Zug0.sucheMaxZiel()* die aktuell größte Zielbahnhofnummer bestimmt und in *MaxZiel* gespeichert. Dann wird mit *Zug0.getAnzahl()* die aktuelle Anzahl der Waggons von *Zug0* ermittelt und in der Variablen *AnzahlWaggons* gespeichert. In der folgenden for-Schleife wird sukzessive ein Waggon von *Zug0* abgekuppelt und in der Hilfsvariablen *einWaggon* zwischengespeichert. Ist die Zielbahnhofsnummer kleiner als das aktuelle *MaxZiel*, so wird er an *Zug2*, andernfalls an *Zug1* angekuppelt. *ZugAufDenAblaufberg(Zug einZug)*: Im Parameter *einZug* wird angegeben, welcher Zug auf den Ablaufberg gebracht wird. Zunächst wird mit *einZug.getAnzahl()* die aktuelle Anzahl der Waggons von *einZug* ermittelt und in der Variablen *AnzahlWaggons* gespeichert. In der folgenden for-Schleife werden alle Waggons nacheinander von *einZug* ab- und an *Zug0* angekuppelt.	4	7	1
3.2	Der Algorithmus beginnt mit einer Schleife, in der geprüft wird, ob *Zug0* noch Waggons hat. Solange das der Fall ist, werden alle Waggons von *Zug0* mit der aktuell größten Zielbahnhofsnummer mittels *WaggonsAbstossen()* an *Zug1* angekuppelt, alle anderen Waggons an *Zug2*. Letztere werden anschließend wieder an *Zug0* angekuppelt. Im nächsten Durchlauf wird die größte Zielbahnhofsnummer der verbliebenen Waggons bestimmt und ebenso verfahren. Der Vorgang ist abgeschlossen, wenn alle Waggons an *Zug1* angekuppelt worden sind und somit *Zug0* leer ist. Die Waggons werden durch den Rangier-Algorithmus nach der Zielbahnhofsnummer sortiert. Zum Schluss werden alle Waggons wieder von *Zug1* an *Zug0* angekuppelt.	4	4	2
4	Mit dem ersten Konstruktor wird ein „leerer“ Zug instanziiert. *Zug1* und *Zug2* verwenden diesen Konstruktor, da sie zu Beginn des Rangiervorgangs noch keine Waggons besitzen. Der zweite Konstruktor erzeugt einen vollständigen Güterzug mit zwölf Waggons. *Zug0* wird mit diesem Konstruktor aufgerufen, um einen ankommenden Zug zu simulieren.		1	3
	Summe 60	**18**	**30**	**12**

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 2013“ vom 20. Juni 2011 zu beachten.

Im Fach Informatik (Grundkurs) werden Vorschläge aus den Kategorien A (Modellierung), B (Datenbanken) und C (theoretische Informatik) vorgelegt, wobei die Prüfungsleistung aus der Bearbeitung von zwei Vorschlägen, einem aus der Kategorie A und einem weiteren aus einer der beiden anderen Kategorien besteht. Es können hierfür insgesamt maximal 100 BE vergeben werden. 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	5	7		**12**
1.2	3	3		**6**
2.1	2	4	2	**8**
2.2		4	4	**8**
3.1	4	7	1	**12**
3.2	4	4	2	**10**
4		1	3	**4**
Summe	**18**	**30**	**12**	**60**

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