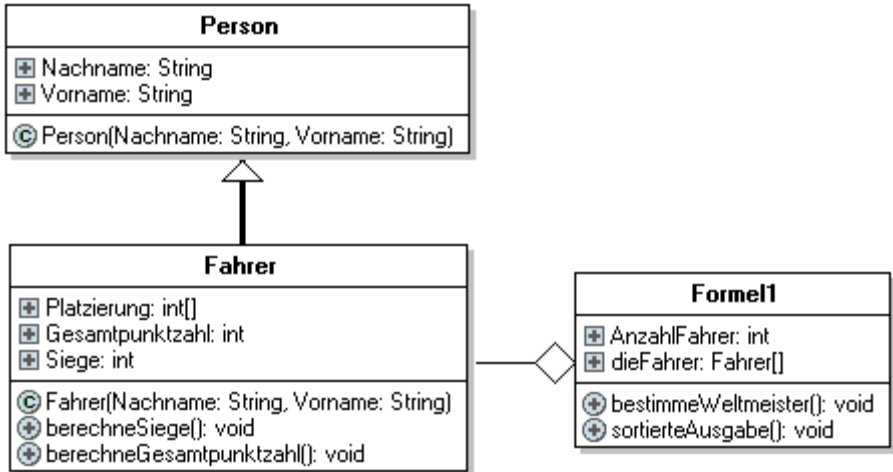


I. Erläuterungen**II. Lösungshinweise und Bewertungsraster**

Entsprechend den Vorgaben der VOGO/BG, Anlage 11 I. Abs. 2.3.1 werden in den nachfolgenden Lösungshinweisen 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.

	Erwartete Leistungen	BE		
		I	II	III
1	Berechnung der Gesamtpunktzahl $8+3+10+10+8+4+8+8+10+10+10+1+6+10+10+5 = 121$	4		
2.1	Die Klasse <i>Person</i> besteht aus den beiden Attributen <i>Nachname</i> und <i>Vorname</i> vom Datentyp <i>String</i> , sowie dem Konstruktor <i>Person</i> . Der Konstruktor hat die beiden Parameter <i>Nachname</i> und <i>Vorname</i> . Mit Hilfe des Konstruktors können neue Personen-Objekte erzeugt werden. Die Klasse <i>Formel1</i> besteht aus den beiden Attributen <i>AnzahlFahrer</i> und <i>dieFahrer</i> sowie den zwei Methoden <i>bestimmeWeltmeister</i> und <i>sortierteAusgabe</i> . Das Attribut <i>dieFahrer</i> ist ein Feld, in dem 22 Fahrer-Objekte gespeichert werden können.	8	8	
2.2	 <pre> classDiagram class Person { +Nachname: String +Vorname: String +Person(Nachname: String, Vorname: String) } class Fahrer { +Platzierung: int[] +Gesamtpunktzahl: int +Siege: int +Fahrer(Nachname: String, Vorname: String) +berechneSiege(): void +berechneGesamtpunktzahl(): void } class Formel1 { +AnzahlFahrer: int +dieFahrer: Fahrer[] +bestimmeWeltmeister(): void +sortierteAusgabe(): void } Person < -- Fahrer Fahrer *-- Formel1 </pre>			
2.3	<pre> public class Fahrer extends Person { int[] Platzierung = new int[18]; int Gesamtpunktzahl; int Siege; public Fahrer(String Nachname, String Vorname) { super(Nachname, Vorname); Gesamtpunktzahl = 0; Siege = 0; for (int i = 0; i < 18; i++) Platzierung[i] = 0; } } </pre>			

3.	<pre> public void berechneSiege() { Siege = 0; for (int i = 0; i < 18; i++) if (Platzierung[i] == 1) Siege++; } public void berechneGesamtpunktzahl() { int Punkte = 0; for (int i = 0; i < 18; i++){ if (Platzierung[i] == 1) Punkte = Punkte + 10; else if (Platzierung[i] == 2) Punkte = Punkte + 8; else if (Platzierung[i] == 3) Punkte = Punkte + 6; else if (Platzierung[i] == 4) Punkte = Punkte + 5; else if (Platzierung[i] == 5) Punkte = Punkte + 4; else if (Platzierung[i] == 6) Punkte = Punkte + 3; else if (Platzierung[i] == 7) Punkte = Punkte + 2; else if (Platzierung[i] == 8) Punkte = Punkte + 1; } Gesamtpunktzahl = Punkte; } </pre>		7	5
4.	<pre> public void bestimmeWeltmeister() { for (int i = 0; i < AnzahlFahrer; i++) { dieFahrer[i].berechneGesamtpunktzahl(); dieFahrer[i].berechneSiege(); } int wm = 0; // Weltmeister for (int i = 1; i < AnzahlFahrer; i++) if (dieFahrer[i].Gesamtpunktzahl > dieFahrer[wm].Gesamtpunktzahl (dieFahrer[i].Gesamtpunktzahl == dieFahrer[wm].Gesamtpunktzahl && dieFahrer[i].Siege > dieFahrer[wm].Siege)) wm = i; System.out.println(dieFahrer[wm].Vorname + " " + dieFahrer[wm].Nachname + " ist Weltmeister!"); } </pre>		7	5
5.	<p>Zuerst sortiert man das Feld, dann gibt man es aus. Als Sortierverfahren kann man Sortieren durch Auswählen benutzen. Dazu teilt man das Feld der Fahrer in einen sortierten und unsortierten Bereich auf. Im unsortierten Bereich sucht man den Fahrer, der die meisten Punkte hat. Dann tauscht man den Fahrer, der am Anfang des unsortierten Bereichs steht, mit dem Fahrer, den man gefunden hat, aus. Dadurch vergrößert sich der sortierte Bereich um einen Fahrer, während der unsortierte Bereich um einen Fahrer abnimmt. Dieses Verfahren wird solange wiederholt, bis der unsortierte Bereich leer ist und somit das Feld der Fahrer sortiert ist.</p> <p>Die Suche des Fahrers mit den meisten Punkten beginnt mit dem ersten Fahrer des unsortierten Bereichs. Man vergleicht nacheinander die Punktzahlen aller Fahrer des unsortierten Bereichs mit dem bisher besten Wert. Hat ein Fahrer eine größere Punktzahl, so wird der bisher beste Wert durch die gefundene größere Punktzahl ersetzt.</p>	8	8	

Sortiert ausgeben			
für i = 1 bis AnzahlFahrer -1			
maxPos = i			
für k = i + 1 bis AnzahlFahrer			
<div style="border: 1px solid black; padding: 5px; text-align: center;"> Fahrer[k].Gesampunktzahl > Fahrer[maxPos].Gesampunktzahl </div>			
J			N
maxPos = k			
Hilfsfahrer = Fahrer[i]			
Fahrer[i] = Fahrer[maxPos]			
Fahrer[maxPos] = Hilfsfahrer			
für i = 1 bis AnzahlFahrer			
Gib Fahrer i aus			

III. Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt gemäß den Bestimmungen in den Anlagen 11 sowie ggf. 9a bis 9e der VOGO/BG in der jeweils gültigen Fassung. Für die Umrechnung von Prozentanteilen der erbrachten Leistungen in Notenpunkte nach §13 Abs. 1 der VOGO/BG gelten die Werte in der Anlage 8 der VOGO/BG in der jeweils gültigen Fassung. Darüber hinaus sind die Vorgaben des Einföhrungserlasses für das Landesabitur 2008 zu beachten.

Im Fach Informatik besteht die Prüfungsleistung aus der Bearbeitung je eines Vorschlags aus den Bereichen Objektorientierte Modellierung und Datenbanken, 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 Teilaufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen

Aufgabe	Bewertungseinheiten in den Anforderungsbereichen			Summe
	AFB I	AFB II	AFB III	
1	4			4
2	8	8		16
3		7	5	12
4		7	5	12
5	8	8		16
Summe	20	30	10	60