

I. Erläuterungen

Aufgabenart

Objektorientierte Modellierung

Voraussetzungen gemäß Lehrplan und Erlass „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2012“ vom 06. Juli 2010

Objektmodell, Klassen

Entwicklung grafischer Benutzeroberflächen

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	Die Benutzeroberfläche enthält Labels, Textfelder und einen Button. Die Textfelder dienen zur Eingabe eines Lottotipps. Mit Labels wird das Formular beschriftet und weitere Labels dienen zur Ausgabe der gezogenen Lottozahlen. Für die Ausgabe des Gewinntextes wird ein weiteres Label benötigt. Mit Hilfe des Buttons wird über die zugehörige Ereignismethode eine Ziehung durchgeführt.	7		
2	Die Klasse <i>Benutzeroberflaeche</i> ist eine Unterklasse der Klasse <i>JFrame</i> . Die Vererbungsbeziehung wird durch <i>Benutzeroberflaeche extends JFrame</i> implementiert. Sie kennt die Klasse <i>Spiel</i> (Assoziation). Die Klasse <i>Spiel</i> kennt (Assoziation) die Klasse <i>Ziehung</i> und hat dazu ein Attribut <i>eineZiehung</i> vom Typ <i>Ziehung</i> . Die Klasse <i>Ziehung</i> kennt die Klasse <i>Trommel</i> und hat dazu ein Attribut <i>dieTrommel</i> vom Typ <i>Trommel</i> . Zwischen den Klassen <i>Trommel</i> und <i>Kugel</i> besteht eine Aggregationsbeziehung, weil eine Trommel 49 Kugeln enthält, welche im Feld <i>dieKugeln</i> gespeichert werden.	4	3	

Aufg.	erwartete Leistungen	BE		
		I	II	III
3	<p>Zuerst werden die Attribute <i>dieKugeln</i> als Feld von Kugeln und <i>Anzahl</i> vom Datentyp <code>int</code> deklariert.</p> <p>Der Konstruktor setzt das Attribut <i>Anzahl</i> auf 49, erzeugt das Feld <i>dieKugeln</i> und belegt es mit 49 von 1 bis 49 durchnummerierten Kugeln.</p> <p>Die Methode <i>ziehen</i> ermittelt eine Zufallszahl von 0 bis <i>Anzahl-1</i> der noch vorhandenen Kugeln und speichert sie in der Variablen <i>pos</i>. Das in dem Feld an der Stelle <i>pos</i> stehende Kugelobjekt wird in der Variablen <i>eineKugel</i> zwischengespeichert. Danach werden alle Kugeln ab <i>pos+1</i> im Feld um einen Speicherplatz nach vorne verschoben und danach die Anzahl der Kugeln um eins verringert. Die Methode liefert das in <i>eineKugel</i> gespeicherte Objekt.</p> <p>Jede Kugel im Feld <i>dieKugeln</i> hat genau eine Nummer. Der beschriebene Algorithmus zieht nur im Feld gespeicherte Kugelobjekte und entfernt nach dem Ziehen die gezogene Kugel aus dem Feld. Eine Kugel kann deshalb kein zweites Mal gezogen werden.</p>	4	8	4
4	<p>Erwartet wird die korrekte Implementierung eines einfachen Sortieralgorithmus entsprechend dem nachstehenden Beispiel.</p> <pre>private void sortieren() { boolean unsortiert = true; int temp; while (unsortiert) { unsortiert = false; for (int i = 0; i < 5; i++) if (dieLottozahlen[i] > dieLottozahlen[i + 1]) { temp = dieLottozahlen[i]; dieLottozahlen[i] = dieLottozahlen[i + 1]; dieLottozahlen[i + 1] = temp; unsortiert = true; } } }</pre> <p>Wie beim Bubblesort-Algorithmus werden in einer Schleife alle paarweise nebeneinander stehenden Zahlen miteinander verglichen. Steht die größere Zahl vor der kleineren, werden beide Zahlen mittels Dreieckstausch getauscht und der Zustand des Feldes auf <i>unsortiert</i> gesetzt. Eine äußere Schleife sorgt dafür, dass der paarweise Vergleich aller Zahlen solange wiederholt wird, bis keine Vertauschung mehr vorgenommen wurde. In diesem Fall ist das Feld dann sortiert.</p>	3	7	
5.1	<p>Die Variable <i>AnzahlRichtige</i> wird mit 0, die boolesche Variable <i>mitZusatzzahl</i> mit <code>false</code> initialisiert. In einer Schleife über alle Tippzahlen wird jede Tippzahl mit allen Lottozahlen und der Zusatzzahl verglichen. Für den Vergleich mit den Lottozahlen wird eine weitere innere Schleife benötigt, innerhalb derer der Vergleich durchgeführt wird. Bei Gleichheit wird <i>AnzahlRichtige</i> inkrementiert. Nach dem Vergleich mit den Lottozahlen muss eine Tippzahl noch mit der Zusatzzahl verglichen werden. Bei Gleichheit wird <i>mitZusatzzahl</i> auf <code>true</code> gesetzt. Nach dem Vergleich der Tippzahlen mit den Lottozahlen und der Zusatzzahl wird das Ergebnis ermittelt. Hat <i>mitZusatzzahl</i> den Wert <code>true</code> und ist <i>AnzahlRichtige</i> mindestens 3 so wird <i>AnzahlRichtige</i> mit <i>Zusatzzahl</i> zurückgegeben, ansonsten nur <i>AnzahlRichtige</i>.</p>		8	2

Aufg.	erwartete Leistungen	BE		
		I	II	III
5.2	<pre> public String gibRichtige() { int AnzahlRichtige = 0; boolean mitZusatzzahl = false; for (int TippzahlNr = 0; TippzahlNr < 6; TippzahlNr++) { for (int LottozahlNr = 0; LottozahlNr < 6; LottozahlNr++) { if (Tipp[TippzahlNr] == eineZiehung.gibLottozahl(LottozahlNr)) AnzahlRichtige++; } if (Tipp[TippzahlNr] == eineZiehung.gibZusatzzahl()) mitZusatzzahl = true; } if (mitZusatzzahl && AnzahlRichtige >= 3) return AnzahlRichtige + " Richtige mit Zusatzzahl"; else return AnzahlRichtige + " Richtige"; } </pre>		4	6
	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 2012“ vom 06. Juli 2010 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	7			7
2	4	3		7
3	4	8	4	16
4	3	7		10
5.1		8	2	10
5.2		4	6	10
Summe	18	30	12	60

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