I. Erläuterungen

Aufgabenart
Objektorientierte Modellierung

Voraussetzungen gemäß Lehrplan und Erlass „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2010“ vom 20. Juni 2008
Standardalgorithmen
rekursive und iterative Verfahren
einfache Such- und Sortierverfahren

II. Lösungshinweise und Bewertungsraster

Den Vorgaben der Anlage 11 I. Abs. 2.3.1 VOGO/BG entsprechend 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.

Aufg.	**erwartete Leistungen**	**BE**		
		I	**II**	**III**
1.1	Es ergibt sich die Reihenfolge: 12 - 18 - 22 - 15 - 19 - 13 22 - 18 - 12 - 15 - 19 - 13 13 - 19 - 15 - 12 - 18 - 22 19 - 13 - 15 - 12 - 18 - 22 18 - 12 - 15 - 13 - 19 - 22 18 - 12 - 15 - 13 - 19 - 22 13 - 15 - 12 - 18 - 19 - 22 15 - 13 - 12 - 18 - 19 - 22 12 - 13 - 15 - 18 - 19 - 22 13 - 12 - 15 - 18 - 19 - 22 12 - 13 - 15 - 18 - 19 - 22	4		
1.2	Die Methode *suche_maxIndex* sucht den Index des größten Pfannkuchens im Bereich *0* bis *obergrenze*, wobei die Obergrenze als Parameter übergeben wird. Die Variablen *maxIndex* und *max* enthalten die Werte für den aktuell größten Pfannkuchen und werden mit *0* und *pfannkuchen[0]* initialisiert. Nun wird das Feld *pfannkuchen* vom 1 bis zur Obergrenze durchlaufen und *pfannkuchen[i]* mit dem aktuell größten Wert *max* verglichen. Ist *pfannkuchen[i]* größer als *max*, so werden der Index i in *maxIndex* und die Größe des i-ten Pfannkuchens in *max* gespeichert. Der so gefundene Index des größten Pfannkuchens wird abschließend als Ergebnis zurückgegeben.	4	4	
	Die Untergrenze *m* und die Obergrenze *n* des zu wendenden Teilfeldes von *pfannkuchen* werden als Parameter an die rekursive Methode *wenden* übergeben. Zunächst wird die Rekursionsbedingung $m < n$ geprüft. Falls diese gilt, so werden mit Hilfe eines Dreieckstausches die Werte der Elemente der Ober- und Untergrenze getauscht. Da nun die beiden äußeren Elemente des Teilfeldes getauscht sind, wird m inkrementiert und n dekrementiert. Mit den neuen Grenzen des verbliebenen Teilfeldes wird dann *wenden2* rekursiv aufgerufen. Die Rekursion terminiert wenn m >= n.	4	5	

Aufg.	erwartete Leistungen	BE		
		I	II	III

1.3	`private void wenden(int m) {`
	`  int anfang = 0;`
	`  int ende = m;`
	`  while (anfang < ende) {`
	`  int hilf = pfannkuchen[anfang];`
	`  pfannkuchen[anfang] = pfannkuchen[ende];`
	`  pfannkuchen[ende] = hilf;`
	`  anfang++;`
	`  ende--;`
	`  }`
	`}`
			8	

2.1	Algorithmus Pfannkuchensortieren für i von (anzahl-1) bis 1 wiederhole max = suche_maxIndex(i) wenden(max) wenden(i)		8	

2.2	`private void sortieren() {`
	`  int maxIndex;`
	`  for (int i = anzahl - 1; i > 0; i--) {`
	`  maxIndex = suche_maxIndex(i);`
	`  wenden(maxIndex);`
	`  wenden(i);`
	`  }`
	`}`
			4	4

3.	`private void sortieren2() {`
	`  int minIndex;`
	`  for (int i = 0; i <= anzahl - 1; i++) {`
	`  minIndex = suche_minIndex(i);`
	`  wenden2(i, minIndex);`
	`  }`
	`}`
			5	4

4	Man sucht im Stapel zunächst den größten Pfannkuchen, schiebt den Wender unter ihn und wendet anschließend den kompletten Teilstapel über dem Wender. Dadurch kommt der größte Pfannkuchen an die oberste Position. Zeigt die nicht verbrannte Seite dieses Pfannkuchens nach oben, so wendet man ihn, so dass die verbrannte Seite oben liegt. Wendet man nun den gesamten Stapel, so gelangt der größte Pfannkuchen ganz nach unten an die gewünschte Position und seine verbrannte Seite zeigt nach unten. Er wird nicht mehr bewegt. Jetzt sucht man den zweitgrößten Pfannkuchen und bringt ihn nach dem gleichen Verfahren an die zweitunterste Position. So fährt man fort, bis der gesamte Stapel sortiert ist.		6	
	Summe 60	**12**	**40**	**8**

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 Erlasses „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2010“ vom 20. Juni 2008 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	4			**4**
1.2	8	9		**17**
1.3		8		**8**
2.1		8		**8**
2.2		4	4	**8**
3		5	4	**9**
4		6		**6**
Summe	**12**	**40**	**8**	**60**

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