

## I. Erläuterungen

### Aufgabenart

materialgebundene Aufgabenstellung


Voraussetzungen gemäß Lehrplan und Erlass „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im Landesabitur 2015“ vom 27. Juni 2013

Q1 objektorientierte Modellierung

Klassendiagramm mit UML, grafische Benutzeroberflächen, iterative Verfahren, einfache Sortieralgorithmen

## 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	<p>Die Klasse <i>Bild</i> hat die Attribute <i>Titel</i> und <i>DatumUhrzeit</i> vom Typ <i>String</i> sowie das Attribut <i>Icon</i> vom Typ <i>ImageIcon</i>. Alle Attribute haben die Sichtbarkeit <i>private</i>.</p> <p>Mit dem Konstruktor wird ein neues <i>Bild</i>-Objekt mit den übergebenen Parametern <i>Titel</i>, <i>DatumUhrzeit</i> und <i>Dateiname</i> angelegt. Mit den öffentlichen <i>get</i>-Methoden können die Werte der Attribute gelesen werden. Außerdem enthält die Klasse die öffentliche Methode <i>liesDatumUhrzeitFormat()</i> mit dem Rückgabotyp <i>String</i>.</p>	8			8
1.2	 <pre> classDiagram     class JFrame     class GUI     class Fotoschau     class Bild     JFrame &lt; -- GUI     GUI --&gt; Fotoschau     Fotoschau o-- Bild           </pre> <p>Die Klasse <i>GUI</i> ist eine Spezialisierung von <i>JFrame</i> und wird durch Vererbung aus der Klasse <i>JFrame</i> abgeleitet. Die Klasse <i>GUI</i> kennt die Klasse <i>Fotoschau</i> (Assoziation). Eine <i>Fotoschau</i> besteht aus mehreren <i>Bildern</i> (Aggregation), die in dem Feld <i>dieBilder</i> gespeichert werden.</p>		8		8
2	Die <i>GUI</i> -Klasse enthält eine <i>Label</i> -Komponente zum Anzeigen des Bildes, einen <i>Button</i> zum Anzeigen des nächsten Bildes und einen zum Sortieren. Außerdem sind zwei <i>Radiobuttons</i> vorhanden, die sicherstellen, dass für eine <i>Fotoschau</i> entweder „normal“ oder „zufällig“ ausgewählt wird. In einer <i>TextField</i> -Komponente werden die <i>Bildinformationen</i> angezeigt.	8			8

Aufg.	erwartete Leistungen	BE			
		I	II	III	Σ
3	<pre> public void nächstesBild() {     if (normal) {         if (Position &lt; AnzahlBilder - 1)             Position++;         else             Position = 0;     } else {         int Zufallszahl;         do {             Zufallszahl = (int) (Math.random() * AnzahlBilder);         } while (Zufallszahl == Position);         Position = Zufallszahl;     } } </pre>	2	6	4	12
4.1	<p>Der Vergleich von Daten des Formats TT.MM.JJJJ hh:mm:ss beginnt mit dem Tag. Beim Sortieren käme z. B. 11.04.2014 00:00:00 nach 07.06.2015 00:00:00, weil 11 größer als 07 ist.</p> <p>Der Vergleich von Daten muss mit dem Jahr beginnen, danach müssen Monat, Tag und Uhrzeit folgen. Dazu muss die Methode <i>liesDatumUhrzeitFormat()</i> einen String im Format JJJJMMTThhmmss zurückgeben.</p>		4		4
4.2	<pre> public String liesDatumUhrzeitFormat() {     String Jahr = DatumUhrzeit.substring(6, 10);     String Monat = DatumUhrzeit.substring(3, 5);     String Tag = DatumUhrzeit.substring(0, 2);     String Zeit = DatumUhrzeit.substring(11, 19);     return Jahr + Monat + Tag + Zeit; } </pre>		3	4	7
4.3	<p>Mit der Variablen <i>i</i> wird das Feld in einen sortierten und in einen unsortierten Bereich geteilt. Der sortierte Bereich besteht am Anfang aus dem ersten Feldelement. Mit der <i>for</i>-Schleife werden die Feldelemente der Reihe nach aus dem unsortierten Bereich entnommen und an der richtigen Stelle im sortierten Bereich eingefügt. Dazu wird zunächst das nächste einzufügende Bild in <i>einBild</i> gespeichert und dessen Datenformat gelesen. In der <i>while</i>-Schleife werden die Datenformate der Bildobjekte mit dem Datenformat des einzufügenden Bildes verglichen. Solange <i>j &gt; 0</i> ist und das Datenformat von <i>dieBilder[j-1]</i> größer ist als das Datenformat von <i>dieBilder[i]</i>, wird das Feldelement mit <i>dieBilder[j] = dieBilder[j-1]</i> um eine Position nach rechts verschoben. Anschließend wird <i>einBild</i> an der Position <i>j</i> gespeichert. Damit wird das aus dem unsortierten Bereich entnommene Element in den sortierten Bereich an der richtigen Stelle eingefügt. (Es handelt sich um den Algorithmus „Sortieren durch Einfügen“.)</p>		9	4	13
	<b>Summe</b>	<b>18</b>	<b>30</b>	<b>12</b>	<b>60</b>

### III. Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt gemäß den Bestimmungen in der OAVO in der jeweils geltenden 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 2015“ vom 27. Juni 2013 zu beachten.

Im Fach Informatik (Grundkurs) können Vorschläge zu den Themen der drei Halbjahre Q1 (objektorientierte Modellierung), Q2 (Datenbanken) und Q3 (Konzepte und Anwendungen der theoretischen Informatik) vorgelegt werden.

Die Prüfungsleistung besteht aus der Bearbeitung von zwei Vorschlägen, einem zum Thema „objektorientierte Modellierung“ und einem weiteren zu einem der beiden anderen Themen. 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.