

Klausur Informatik Q1, RH, 5.10.2020, Erwartungshorizont

Aufgabe 1: Vererbung (AFB I-II)

<p>a) An welcher (einen) Stelle ist die Vererbungshierarchie inhaltlich falsch modelliert? Wie wäre sie korrekt?</p> <p>Das Motorrad ist kein Automobil und muss direkt von Fortbewegungsmittel abgeleitet werden.</p>	4	
<p>b) Erläutere die Grundlagen der Vererbung (Schwerpunkt: Abstraktion und Spezialisierung) an diesem Beispiel.</p> <p>Bei der Vererbung gibt es Elternklassen (Superklassen, Oberklassen) und Kindklassen (Unterklassen). Die Unterklassen erben, je nach Zugriffsregelung, alle Variablen und Methoden der Oberklasse, so dass diese verwendet werden können.</p> <p>Bei der Abstraktion werden gemeinsame Elemente identifiziert, die in eine Superklasse ausgelagert werden. So haben sowohl Fahrrad, Pferd als auch Automobil die Eigenschaften Geschwindigkeit und Reichweite, die in die Oberklasse Fortbewegungsmittel eingebracht werden.</p> <p>Bei der Spezialisierung gibt es eine Erweiterung der Eigenschaften in der Vererbungshierarchie nach unten. So werden die Unterklassen spezifischer bzw. spezieller. Die Unterklasse Automobil hat z.B. die Eigenschaft Kofferraumvolumen, die die Klasse Fahrrad nicht braucht. Hingegen braucht die Klasse Fahrrad die Ritzelanzahlen vorne und hinten.</p>	8	
	12	

c) Codiere die Datenklassen Fortbewegungsmittel. Füge auch Setter und Getter hinzu.

<pre>public class Fortbewegungsmittel {</pre>	1	
<pre> int geschwindigkeit; int reichweite;</pre>	4	
<pre> public Fortbewegungsmittel() { } </pre>	1	
<pre> public void setGeschwindigkeit(int g){ geschwindigkeit = g; } public void setReicchweite(int r){ reichweite = r; } </pre>	6	
<pre> public int getGeschwindigkeit(){ return geschwindigkeit; } public int getReichweite(){ return reichweite; } } </pre>	6	
	18	

Klausur Informatik Q1, RH, 5.10.2020, Erwartungshorizont

d) Codiere alle anderen Klassen (in der korrigierten Version) ohne Setter und Getter.

<pre>public class automobil extends Fortbewegungsmittel {</pre>	1	
<pre> int baujahr; String hersteller; int leistung; int kofferraumvolumen;</pre>	4	
<pre> public automobil() { super(); } }</pre>	1	
<pre>public class Pferd extends Fortbewegungsmittel {</pre>	1	
<pre> int geburtsjahr; String rasse;</pre>	2	
<pre> public Pferd() { super(); } }</pre>	1	
<pre>public class Fahrrad extends Fortbewegungsmittel {</pre>	1	
<pre> int baujahr; String hersteller; String kategorie; int[] ritzelVorne; int[] ritzelHinten;</pre>	4	
<pre> public Fahrrad() { super(); } }</pre>	1	
<pre>public class PKW extends Automobil {</pre>	1	
<pre> String kategorie;</pre>	1	
<pre> public PKW() { super(); } }</pre>	1	
<pre>public class Motorrad extends Fortbewegungsmittel {</pre>	1	
<pre> String kategorie;</pre>	1	
<pre> public Motorrad() { super(); } }</pre>	1	
	22	

Klausur Informatik Q1, RH, 5.10.2020, Erwartungshorizont

e) *Schreibe für die Klasse Fahrrad eine strukturierte Ausgabemethode für die Konsole. (inkl. Zusatzpunkte)*

Da für die Ausgabe der Arrays noch keine Übung im Unterricht erfolgte, werden diese Strukturen nur als Zusatzpunkte gewertet. Außerdem wird auf die Einschränkung verzichtet, dass die Variablen eigentlich nicht direkt auslesbar sind, wenn sie private gesetzt wurden.		
<code>public void ausgabe() {</code>	1	
<code>System.out.println("Hersteller: "+hersteller);</code> <code>System.out.println("Baujahr: "+baujahr);</code> <code>System.out.println("Kategorie: "+kategorie);</code> <code>System.out.println("Baujahr: "+baujahr);</code>	4	
<code>System.out.println("Geschwindigkeit: "+getGeschwindigkeit());</code> <code>System.out.println("Reichweite: "+getReichweite());</code>	4	
Zusatzpunkte: <code>for(int i = 0; i<ritzelVorne.length; i++){</code> <code>System.out.print(ritzelVorne[i]);</code> <code>}</code> Analog für ritzelHinten.		
	9	

f) *Was bedeutet der Befehl super()? Was ist mit „overriding“ gemeint?*

Der super-Befehl ruft den Konstruktor der Oberklasse auf. Er muss dieselben Übergabeparameter haben wie dieser. Overriding meint das Überschreiben einer Methode bei der Vererbung. Soll eine Unterklasse eine abweichende Methode gleichen Namens erhalten, so wird diese programmiert und gilt dann für diese Klasse und alle weiteren Unterklassen.	4	
---	---	--

Aufgabe 2: Suchen, suchen, suchen... (AFB I-II-III)

Gegeben sei ein Array bestehend aus Integer-Elementen (Ganzzahlen).

a) <i>Gibt es für die lineare und die binäre Suche unterschiedliche Voraussetzungen für die Gestalt des Arrays?</i> Das Array muss für die binäre Suche sortiert sein, denn auf unsortierten Arrays funktioniert das Teilen nicht. Die lineare Suche braucht diese Voraussetzung nicht, d.h. es kann sowohl in unsortierten wie sortierten Arrays gesucht werden.	4	
b) <i>Erläutere den Unterschied der Verfahren an einem selbst gewählten Beispiel.</i> 5 7 11 13 17 21 23 Gesucht sei die 21. Die lineare Suche beginnt mit dem ersten Element 5 und vergleicht mit dem gesuchten Element. Bei der linearen Suche wird dann Schritt für Schritt weitergegangen, also mit 7, 11, 13, 17, und letztlich 21. Immer wenn das gesuchte Element nicht gefunden wurde, wird der Suchindex um Eins hochgesetzt. Die binäre Suche hingegen nimmt nicht das erste, sondern das mittlere Element 13. Danach muss bei der binären Suche nur noch in einem der Bereiche gesucht werden. Ist nämlich der gesuchte Wert größer als der mittlere, so findet sich das gesuchte Element wegen der Sortierung in der rechten Hälfte der Reihung. Im anderen Fall ist er der Wert selbst oder findet sich in der linken Hälfte. Da $21 > 13$ wird in der rechten Hälfte weitergesucht. Dazu wird die neue Mitte bestimmt, hier die 21 selbst (wenn aufgerundet wird). Da $21 = 21$ bricht der Suchalgorithmus ab. Wäre das Element nicht gefunden worden, so wäre das Prinzip der Halbierung weitergeführt worden.	10	
c) <i>Stelle Überlegungen zur Effizienz der Verfahren an: Was müsste gezählt werden, um die Verfahren zu vergleichen? Wie groß wären diese Zahlenwerte für Dein Beispiel?</i> Man könnte die Anzahl der Vergleiche bestimmen. Für die lineare Suche wären das im obigen Beispiel 6, für die binäre Suche nur 2.	5	
	19	

Klausur Informatik Q1, RH, 5.10.2020, Erwartungshorizont

Aufgabe 3: Algorithmen analysieren (AFB II-III)

a) Durchlaufe den Algorithmus mit den Eingaben 4 und 8.

m	n	r
4	8	$8 - 4 = 4$
4	4	$4 - 4 = 0$
4	0	Abbruch

b) Durchlaufe den Algorithmus mit den Eingaben 64 und 24.

m	n	r
64	24	$64 - 24 = 40$
40	24	$40 - 24 = 16$
24	16	$24 - 16 = 8$
16	8	$16 - 8 = 8$
8	8	$8 - 8 = 0$
8	0	Abbruch

c) Durchlaufe den Algorithmus mit den Eingaben 11 und 65.

m	n	r
11	65	11
65	11	$65 - 11 = 54$
11	54	11
54	11	$54 - 11 = 43$
11	43	11
43	11	$43 - 11 = 32$
11	32	11
32	11	$32 - 11 = 21$
11	21	11
21	11	$21 - 11 = 10$
11	10	$11 - 10 = 1$
10	1	$10 - 1 = 9$
1	9	1
9	1	$9 - 1 = 8$
1	8	1
8	1	$8 - 1 = 7$
1	7	1
7	1	$7 - 1 = 6$
1	6	1
6	1	$6 - 1 = 5$
1	5	1
5	1	$5 - 1 = 4$
1	4	1
4	1	$4 - 1 = 3$
1	3	1
3	1	$3 - 1 = 2$
1	2	1
2	1	$2 - 1 = 1$
1	1	$1 - 1 = 0$
		Abbruch

d) Was findet der Algorithmus mathematisch ausgedrückt?

Der Algorithmus bestimmt den größten gemeinsamen Teiler.

a)	4	
b)	6	
c)	6	
d)	4	
	20	

Punkteübersicht

A1 a)	b)	c)	d)	e)	A2 a)	b)	c)	A4 a)	b)	c)	d)	Summe
4	8	18	22	9	4	10	5	4	6	6	4	100