

Klausur
Einführung in die Programmierung
19.02.2018 (WS 2017/18)

Hinweise

1. Bearbeitungszeit: 120 Minuten
2. Zum Bestehen der Klausur sind 50 Punkte erforderlich.
3. Bitte auf das erste Blatt Name, Studiengang und Matrikelnummer eintragen.
4. Erlaubte Hilfsmittel: keine
5. Die Klammerung der Klausur nicht lösen!
6. Die Lösungen in die Klausurvorlage eintragen. Weiteres Papier können Sie bei der Klausuraufsicht anfordern.
7. Alle vorgegebenen und zu erstellenden Programmtexte beziehen sich auf die Programmiersprache Java.

Viel Erfolg !

Aufgabe	Max. Punktzahl	Erreichte Punktzahl
1 (Multiple Choice, Wissen)	10	
2 (Ein-/Ausgabe, Kontrollstrukturen)	13	
3 (Programmieraufgaben)	12	
4 (Array-Programmierung)	13	
5 (Klassen und Objekte)	11	
6 (UML nach Java übertragen)	20	
7 (Speicherabbild)	10	
8 (Programmausgabe)	11	
<i>Bonuspunkte</i>	<i>10</i>	
Gesamtpunktzahl	100+10=110	

Aufgabe 1 (Multiple Choice, Wissen)**[10 Punkte]**

- a) Kreuzen Sie in den folgenden Teilaufgaben an, welche Aussagen richtig und welche falsch sind. Für jede korrekt markierte Aussage erhalten Sie 0,5 Punkte, für jede falsch markierte werden 0,5 Punkte abgezogen. Pro Teilaufgabe erhalten Sie mindestens 0 Punkte.

Bewerten Sie folgende Aussagen zu Zahlendarstellungen	Richtig	Falsch
Die Binärdarstellung der Dezimalzahl 106 ist 111010.		
Die Hexadezimaldarstellung der Dezimalzahl 106 ist 6A.		
Die Binärdarstellung der Dezimalzahl 106 benötigt 7 Stellen.		
Die Hexadezimalzahl A6 entspricht der Dezimalzahl 167.		

Welche der folgenden Zeichenketten dürfen in einem Java-Programm als Bezeichner benutzt werden?	Richtig	Falsch
Snoke		
Jabba		
R2-D2		
Leia Organa		

Bewerten Sie folgende Aussagen. Ein Objektattribut mit Paketsichtbarkeit	Richtig	Falsch
wird in UML unterstrichen.		
wird in UML durch ~ gekennzeichnet.		
wird in Java durch <code>static</code> gekennzeichnet.		
wird in Java durch <code>package</code> gekennzeichnet.		

Bewerten Sie die folgenden Aussagen. Sei A eine abstrakte Klasse in Java. Dann gilt:	Richtig	Falsch
A darf keinen Konstruktor besitzen.		
Von A können keine Objekte erzeugt werden.		
Von A dürfen keine Unterklassen gebildet werden.		
A darf nur abstrakte Methoden besitzen.		

- b) Was bedeutet Mehrfachvererbung?

Aufgabe 2 (Ein-/Ausgabe, Kontrollstrukturen)**[13 Punkte]**

Ergänzen Sie die main-Methode der Klasse Auswahl so, dass zunächst 9 Ganzzahlen (int) von der Konsole eingelesen werden und anschließend das folgende Auswahlmenü auf der Konsole angezeigt wird:

- (1) Zeige die 1. bis 7. eingegebene Zahl
- (2) Zeige die achte eingegebene Zahl
- (3) Zeige die neunte eingegebene Zahl
- (4) Beende das Programm

Nach Ausgabe des Menüs sollen so lange Zahlen zwischen 1 und 4 von der Konsole eingelesen und die zu der jeweils ausgewählten Option angegebene(n) Zahl(e)n ausgegeben werden, bis der Benutzer eine 4 eingibt. Die Eingabe der 9 Zahlen und die Ausgabe des Menüs sollen nur einmal erfolgen.

Geben Sie notwendige import-Anweisungen an. **Benutzen Sie keine if- oder if-else-Abfragen.**

```
public class Auswahl
{
    public static void zeigeMenue()
    {
        System.out.println("(1) Zeige die 1. bis 7. eingegebene Zahl");
        System.out.println("(2) Zeige die achte eingegebene Zahl");
        System.out.println("(3) Zeige die neunte eingegebene Zahl");
        System.out.println("(4) Beende das Programm");
    }

    public static void main(String[] args)
    {
```

}

}

Aufgabe 3 (Programmieraufgaben)**[12 Punkte]**

- a) Teilt eine ganze Zahl $t > 0$ eine weitere ganze Zahl $n > 0$ ganzzahlig, so nennt man t Teiler von n . Beispiele: 2 ist ein Teiler von 6, 4 ist aber kein Teiler von 6. Schreiben Sie eine Methode `teileranzahl`, die zu gegebener Zahl $n > 0$ die Anzahl der Teiler von n (von 1 und n) bestimmt.

Beispiel: 6 hat die Teiler 1, 2, 3 und 6, also ist die Anzahl der Teiler 4.

Hinweis:

Zur Lösung dürfen Sie keine Methoden aus der Java-Klassenbibliothek verwenden.

```
public static int teileranzahl(int n)
{
    //Annahme:  $n > 0$ 
```

```
}
```

- b) Eine Zahl $p > 0$ ist eine Primzahl, wenn sie genau 2 Teiler besitzt. Jede Zahl $n > 0$ lässt sich als Produkt von Primzahlen darstellen, z.B. $36 = 2 \cdot 2 \cdot 3 \cdot 3$.

Schreiben Sie eine Methode `primzahlzerlegung`, die zu gegebener Zahl n diese Primzahlzerlegung als String zurückgibt, d.h. für den konkreten Parameter n soll n als Produkt von Primzahlen dargestellt werden. Bsp.: Ist der konkrete Parameter 36, so soll der String „ $2 \cdot 2 \cdot 3 \cdot 3$ “ zurückgegeben werden.

Hinweis:

Zur Lösung dürfen Sie keine Methoden aus der Java-Klassenbibliothek verwenden. Sie dürfen aber die Methode `teileranzahl` aus Aufgabenteil a) verwenden.

```
public static String primzahlzerlegung(int n)
{
    //Annahme: n>0
```

```
}
```

Aufgabe 4 (Array-Programmierung)**[13 Punkte]**

- a) Die Transponierte eines rechteckigen, zweidimensionalen Arrays ist das Array, das durch Vertauschen von Zeilen und Spalten entsteht.

Beispiel:

1	2	3	4
5	6	7	8
9	10	11	12

Array

1	5	9
2	6	10
3	7	11
4	8	12

Transponierte

Aufgabe: Implementieren Sie die Methode

`int[][] berechneTransponierte(int[][] feld)`,
welche die Transponierte des Arrays `feld` erzeugt, mit den korrekten Werten füllt und
als Ergebnis zurückgibt.

```
public static int[][] berechneTransponierte(int[][] feld)
{
    //Annahme: feld ist ein rechteckiges Array ungleich null
```

```
}
```

- b) Ein rechteckiges, zweidimensionales Array ist symmetrisch, wenn das Array und die Transponierte gleich sind.

Beispiel:

1	2	3	4
2	1	5	5
3	5	9	9
4	5	9	8

Array

1	2	3	4
2	1	5	5
3	5	9	9
4	5	9	8

Transponierte

symmetrisches Array

1	2	3	4
2	1	5	5
1	5	9	9
1	5	9	8

Array

1	2	1	1
2	1	5	5
3	5	9	9
4	5	9	8

Transponierte

kein symmetrisches Array

Aufgabe: Implementieren Sie die Methode

boolean istSymmetrisch(**int**[][] feld),

die überprüft, ob das Array feld symmetrisch ist. Ist feld symmetrisch, so soll true, sonst false zurückgegeben werden.

Hinweis: Zur Lösung dürfen Sie die Methode berechneTransponierte aus a) benutzen.

```
public static boolean istSymmetrisch(int[][] feld)
```

```
{
```

```
    //Annahme: feld ist ein rechteckiges Array ungleich null
```

```
}
```

Aufgabe 5 (Klassen und Objekte)**[11 Punkte]**

Betrachten Sie die folgenden Klassen:

```
public class Test {  
    public static void main(String[] args)  
    {  
        Person p1 = new Person("Rolf", 60);  
        Person p2 = new Person("Gudrun", 41);  
        // Zeitpunkt a)  
        // Anfang Aufgabenteil b)
```

```
        // Ende Aufgabenteil b)  
        // Zeitpunkt b)  
        System.out.println(porsche);  
        System.out.println(vwkaefer);  
    }  
}
```

```
public class Person  
{
```

```
    private String name;  
    private int alter;  
    private int id;
```

```
    _____ // Aufgabenteil a)
```

```
    public Person(String name, int alter)  
    {  
        this.name = name;  
        this.alter = alter;
```

```
    _____ // Aufgabenteil a)
```

```
    }
```

```
    // Aufgabenteil c)
```

```
}
```

```

public class Kfz {
    private String bezeichnung;
    private Person besitzer;

    public Kfz(String bezeichnung, Person besitzer)
    {
        this.bezeichnung = bezeichnung;
        this.besitzer = besitzer;
    }
    // Aufgabenteil c)

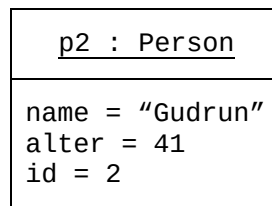
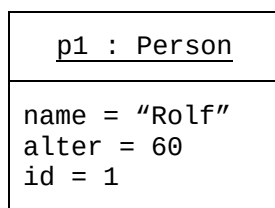
```

```

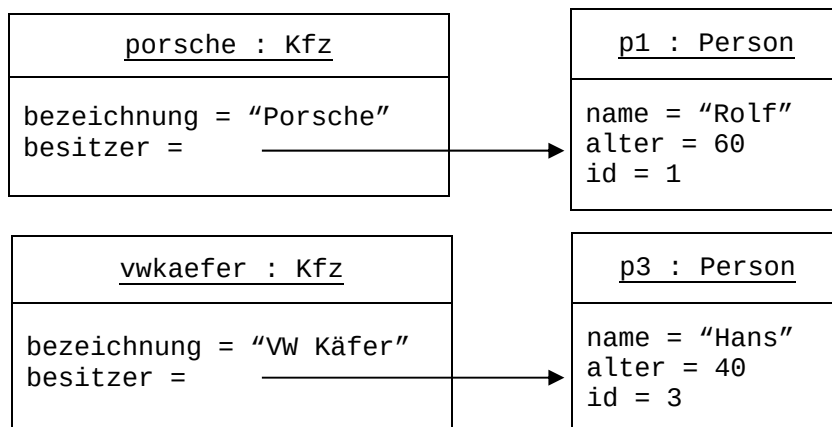
}

```

- a) Ergänzen Sie die Klasse Person um ein Attribut und modifizieren Sie den Konstruktor dieser Klasse so, dass zum Zeitpunkt a) das folgende UML-Diagramm entstanden ist. Tragen Sie die Modifikationen in den obigen Programmtext ein.



- b) Ergänzen Sie die main-Methode im angegebenen Bereich so, dass zum Zeitpunkt b) die folgenden Objekte existieren:



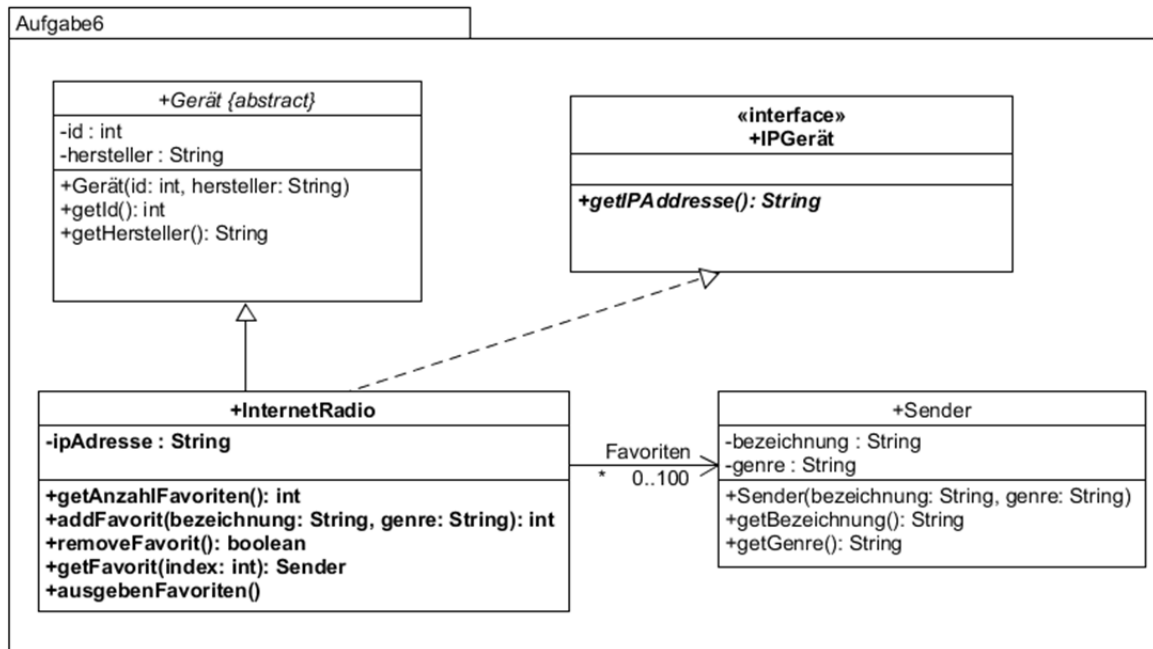
- c) Ergänzen Sie die Klassen Kfz und Person um jeweils eine Methode, so dass die Ausgabe der main-Methode wie folgt aussieht:
- ```

Porsche mit Besitzer Rolf, 60 Jahre alt
VW Käfer mit Besitzer Hans, 40 Jahre alt

```

**Aufgabe 6 (UML nach Java übertragen)****[20 Punkte]**

Gegeben sei das folgende UML-Klassendiagramm. Programmieren Sie die Schnittstelle IP-Gerät und die Klasse InternetRadio vollständig in Java.

Hinweis:

Außer String dürfen keine Klassen der Java-Klassenbibliothek verwendet werden!

Ergänzen Sie die Klasse InternetRadio um

- Attribute zur Verwaltung der Assoziation
- einen sinnvollen Konstruktor
- weitere Methoden, soweit erforderlich

Die Methoden sollen folgendes leisten:

- `getAnzahlFavoriten`: gibt die Anzahl der Sender in der Favoritenliste zurück
- `addFavorit`: fügt einen Sender an einer freien Position ein. Falls eingefügt werden konnte, ist der Rückgabewert die Position, an der eingefügt wurde, sonst -1.
- `removeFavorit`: entfernt den ersten Eintrag in der Favoritenliste. Dabei darf keine Lücke entstehen. Der Rückgabewert ist `true`, falls ein Eintrag entfernt wurde, ansonsten `false`.
- `getFavorit`: gibt den Sender an der Position `index` in der Favoritenliste zurück, falls vorhanden, ansonsten `null`.
- `ausgebenFavoriten`: gibt für die gespeicherten Sender die Bezeichnung und das Genre auf der Konsole aus.





**Aufgabe 7 (Speicherabbild)****[10 Punkte]**

Zeichnen Sie das Speicherabbild zu folgendem Programmcode zum Zeitpunkt 1, d.h. nach der letzten Anweisung und vor Verlassen des Rumpfs der Methode methode in der Klasse A.

```
public class Test {
 public static void main
 (String[] args)
 {
 A a = new A(3);
 A b = new B(1,4);
 b.methode(6);
 }
}

public class A {
 private static int c = 1;
 private int m;
 private int n;

 public A(int n)
 {
 m = 5;
 c += m+n;
 }
 public void methode(int i)
 {
 m -= i;
 erhoehe(i);
 // Zeitpunkt 1

 }
 public void erhoehe(int i)
 {
 m += 2*i;
 }
}

public class B extends A {
 private int m;

 public B(int m, int n)
 {
 super(n);
 this.m = m + 1;
 }
 public void erhoehe(int i)
 {
 m += i;
 super.erhoehe(i);
 }
}
```

args

null

main

**Aufgabe 8 (Programmausgabe)****[11 Punkte]**

Geben Sie an, welche Ausgaben dieses Programm produziert.

```
public class Test
{
 public static void main(String[] args)
 {
 A einA = new A();
 System.out.println("A: " + einA.x);
 einA.print();

 A einB = new B();
 System.out.println("B: " + einB.x);
 einB.print();

 C einC = new C();
 einC.print();
 System.out.println("C: " + einC.x);
 }
}
```

```
public class A
{
 public static int z;
 public int x = 10;

 public A()
 {
 x++;
 z++;
 }
 public void print()
 {
 System.out.println("A: "+ ++x);
 System.out.println("A: "+ ++z);
 }
}
```

```
public class B extends A
{
 public int x = 100;

 public B()
 {
 x++;
 z++;
 }

 public void print()
 {
 System.out.println("B: "+ x);
 System.out.println("B: "+ ++z);
 }
}
```

```
public class C extends B
{
 public int x = 1000;

 public void print()
 {
 super.print();
 System.out.println("C: "+ x);
 System.out.println("C: "+ ++z);
 }
}
```

Ausgaben: