

Klausur
Einführung in die Programmierung
18.07.2018 (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)	14	
3 (Programmieraufgaben)	12	
4 (Array-Programmierung)	13	
5 (Klassen und Objekte)	11	
6 (UML nach Java übertragen)	20	
7 (Speicherabbild)	10	
8 (Programmausgabe)	10	
<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 die Korrektheit der folgenden Anweisungsfolgen.	Richtig	Falsch
<code>double x; float y = x; x = y;</code>		
<code>long x; float y = x; x = y;</code>		
<code>long x = true;</code>		
<code>int x=5; long y=5; boolean z = (x==y);</code>		

Bewerten Sie folgende Aussagen.	Richtig	Falsch
Ein Quellprogramm für Java wird vom Java-Compiler in Maschinencode übersetzt.		
Ein Quellprogramm für Java wird durch einen Interpreter ausgeführt.		
Ein Quellprogramm für Java besteht immer aus einer Datei.		
Java-Bytecode wird von einer virtuellen Maschine ausgeführt.		

Welche der folgenden Java-Ausdrücke werten zu true aus, welche zu false? (Die Einfärbung soll nur das Zählen der Zeichen erleichtern!)	true	false
<code>0xffff + 1 == 0xffff1</code>		
<code>0xfffffff + 1 == 0</code>		
<code>0x0fffff + 1 < 0</code>		
<code>0xfffffffL + 1 < 0</code>		

Bewerten Sie die folgenden Aussagen. Sei A ein Interface in Java. Dann gilt:	Richtig	Falsch
A muss einen Konstruktor besitzen.		
Von A können keine Objekte erzeugt werden.		
Von A dürfen keine Unterklassen gebildet werden.		
A darf keine privaten Methoden enthalten.		

- b) Erläutern Sie die Zweierkomplementdarstellung des Datentyps `int` in Java. Geben Sie als Beispiel je eine positive und eine negative Zahl in dieser Darstellung an.

Aufgabe 2 (Ein-/Ausgabe, Kontrollstrukturen)**[14 Punkte]**

In dieser Aufgabe sollen zunächst die Ergebnisse von 3 WM-Spielen eingelesen und die daraus resultierende Tabelle ausgegeben werden. Für einen Sieg werden 3 Punkte für den Sieger vergeben, bei einem Unentschieden bekommt jede der beiden Mannschaften 1 Punkt. Die teilnehmenden Mannschaften sind Italien, Niederlande und Deutschland, wobei jede dieser Mannschaften gegen jede andere genau einmal spielt.

Zunächst soll die Zeile

(1) Italien – Niederlande

ausgegeben und zwei Zahlen (int) nacheinander von der Konsole eingelesen werden, die das Ergebnis der Partie ergeben. Werden z.B. die Zahlen 4 und 1 eingegeben, so hat Italien mit 4 zu 1 gewonnen.

In gleicher Weise sollen dann auch die Ergebnisse der Partien Niederlande – Deutschland und Deutschland – Italien eingelesen werden.

Nach Einlesen der Ergebnisse soll dann die Tabelle in der Form

Mannschaft Punkte
ausgegeben werden, z.B.

Italien 4

Niederlande 2

Deutschland 1

Die Tabelle muss hierbei **nicht** nach Punkten sortiert ausgegeben werden!

Haben zwei Mannschaften die gleiche Punktezahl, so soll die gesamte Eingabe der Ergebnisse wiederholt werden, bis sich eine eindeutige Tabelle ergibt.

Ergänzen Sie die main-Methode der Klasse Spielplan entsprechend.

Geben Sie notwendige import-Anweisungen an. Sie dürfen eigenständig Methoden ergänzen, wenn Sie diese benötigen.

```
public class Spielplan
{
    public static void main(String[] args)
    {
        int punkteDeutschland = 0;
        int punkteItalien = 0;
        int punkteNiederlande = 0;
```

}

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

- a) In dieser Teilaufgabe sollen die in einem eindimensionalen int-Array abgespeicherten Zahlen aufsummiert werden.

Beispiel: Ist das Feld mit den Zahlen 5, 1, 5, 4 gegeben, so soll die Summe $5+1+5+4 = 15$ berechnet und zurückgegeben werden (keine Ausgabe auf der Konsole).

Hinweis:

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

```
public static int berechneSumme(int[] feld)
{
```

```
}
```

- b) Nun soll aus einer ganzen Zahl $n \geq 0$ ein int-Array erzeugt werden, dass die Ziffern der Dezimaldarstellung von n enthält.
Beispiel: Für die Zahl 5145 soll das Feld der Länge 4 mit den Zahlen 5, 1, 4, 5 zurückgegeben werden.

Hinweis:

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

Sie dürfen aber die folgende Methode zur Bestimmung der Anzahl der Ziffern in der Dezimaldarstellung einer Zahl $n \geq 0$ nutzen:

```
public static int laengeDezimalDarstellung(int n)
{
    //Annahme:  $n \geq 0$ 

    int laenge = 1;
    while (n > 9) {
        laenge++;
        n = n / 10;
    }
    return laenge;
}
```

```
public static int[] dezimalDarstellung(int n)
{
    //Annahme: n>=0
```

```
}
```

- c) Geben Sie nun eine Methode an, die für ganze Zahlen $n \geq 0$ die Quersumme berechnet. Beispiel: Für die Zahl 5145 soll $5+1+4+5 = 15$ zurückgegeben werden.

Hinweis:

Zur Lösung dürfen Sie keine Methoden aus der Java-Klassenbibliothek verwenden. Sie dürfen aber die Methoden aus den Teilaufgaben a) und b) nutzen.

```
public static int quersumme(int n)
{
    //Annahme: n>=0
```

```
}
```

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

- a) Digitale Bilder lassen sich als rechteckige, zweidimensionale Arrays von Bytes speichern, wobei 0 schwarz und 255 weiß entspricht.

Ein Bild ist horizontal symmetrisch, wenn in jeder Zeile das erste und das letzte, das zweite und das vorletzte, usw. Bildelement (Arrayelement) den gleichen Wert hat.

Beispiele:

0	10	10	0
235	69	69	235
120	75	75	120

Horizontal symmetrische Bilder

0	40	0
235	80	235
128	99	128

0	10	10	0
235	210	99	43
17	93	128	167

Kein horizontal symmetrisches Bild

Aufgabe: Implementieren Sie die Methode

boolean istHorizontalSymmetrisch(**byte**[][] bild),
welche true zurückgibt, wenn das Bild horizontal symmetrisch ist

```
public static boolean istHorizontalSymmetrisch(byte[][] bild)
{
    //Annahme: bild ist ein rechteckiges Array mit Seitenlängen
    //          größer 0
}
```

- b) Eine wichtige Operation zur Bildbearbeitung ist das Drehen eines Bildes um 90 Grad nach links.

Beispiel:

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

Bild

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

gedrehtes Bild

Aufgabe: Implementieren Sie die Methode

byte[][] drehe90GradLinks(byte[][] bild),
welche ein neues Array erzeugt, mit den korrekten Werten füllt und als Ergebnis zurückgibt.

```
public static byte[][] drehe90GradLinks(byte[][] bild)
{
    //Annahme: bild ist ein rechteckiges Array mit Seitenlängen
    //          größer 0
```

```
}
```

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

Betrachten Sie die folgenden Klassen:

```
public class ZooTest {  
    public static void main(String[] args) {  
        // Aufgabenteil b)
```

```
        // Zeitpunkt 1
```

```
        System.out.println(t1);  
        System.out.println(t2);
```

```
    }  
}
```

```
public interface Freund {  
    String getName();  
}
```

```
public abstract class Tier {  
    protected String name;  
    private Freund freund;  
  
    public Tier(String name, Freund freund) {  
        this.name = name;  
        this.freund = freund;  
    }  
    public abstract String gibLaut();  
  
    // Aufgabenteil c)
```

```
}
```

```
public class Ziege extends Tier {  
    public Ziege(String name, Freund freund){  
        super(name, freund);  
    }  
  
    public String gibLaut() {  
        return "Mäh";  
    }  
}
```

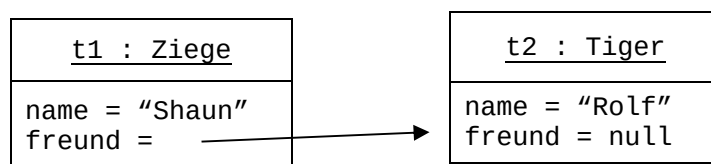
```
// Aufgabenteil a) + c)
```

```
public class Tiger extends Tier _____ {
```

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There are no vertical margin lines, and the paper is completely blank except for the lines themselves.

}

- Ergänzen Sie die Klasse Tiger (Kopf und notwendige Methoden).
- Ergänzen Sie nun die main-Methode so, dass zum Zeitpunkt 1 das folgende UML-Diagramm entstanden ist.



- c) Ergänzen Sie die Klasse `Tier` um eine Methode, so dass die Ausgabe der `main`-Methode wie folgt aussieht:

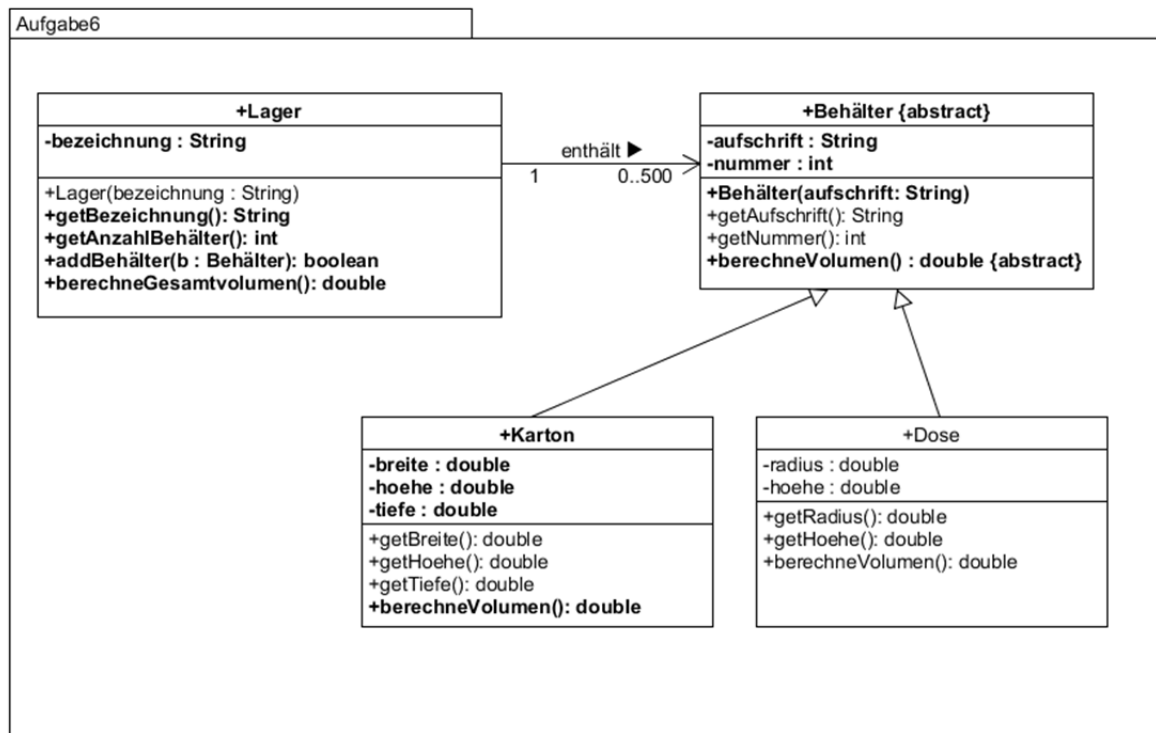
Shaun: Mäh!

Rolf: Grrr!

Falls nötig, passen Sie die entsprechende Methode der Klasse Tiger an.

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

Gegeben sei das folgende UML-Klassendiagramm. Programmieren Sie die Klassen Lager, Behälter und Karton in Java entsprechend den unten aufgeführten Erläuterungen. Programmieren Sie von den Methoden nur diejenigen, die **fettgedruckt** sind.



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

Erläuterungen:

Klasse Lager

- Ergänzen Sie die Attribute zur Verwaltung der Assoziation.
- Die Methode `getAnzahlBehälter` gibt die Anzahl der Behälter des Lagers zurück.
- Die Methode `addBehälter` fügt einen Behälter ein. Falls eingefügt werden konnte, ist der Rückgabewert **true**, ansonsten **false**.
- Die Methode `berechneGesamtVolumen` berechnet die Summe der Volumina aller Behälter.

Klasse Behälter

- Beim Erzeugen eines Behälters soll die Nummer automatisch vergeben werden. Der erste Behälter hat die Nummer 1. Ergänzen Sie dafür notwendige Attribute.

Klasse Karton

- Ergänzen Sie die Klasse um einen sinnvollen Konstruktor.
- Die Methode `berechneVolumen` soll das Volumen aus den Attributwerten berechnen.

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

Zeichnen Sie das Speicherabbild zu folgendem Programmcode zum Zeitpunkt 1, d.h. vor der return-Anweisung der Funktion `summe` in der Klasse `A`.

```
public class Test {
    public static void main
        (String[] args)
    {
        int n;
        A a = new A(1,2);
        A b = new B(3,4);
        n = b.summe(10);
        System.out.print(n);
    }
}

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

    public A(int m, int n)
    {
        this.m = m;
        this.n = n;
        c++;
    }
    public int summe(int i)
    {
        int erg;
        erg = m + n + i;
        // Zeitpunkt 1
        return erg;
    }
}

public class B extends A {
    private int m;

    public B(int m, int n)
    {
        super(n, m);
        this.m = m + n;
    }
    public int summe(int i)
    {
        int erg = m;
        erg += super.summe(i);
        return erg;
    }
}
```

args

null

main

Aufgabe 8 (Programmausgabe)**[10 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(20);
        einA.print(2.5);

        A einB = new B();
        System.out.println("B: " + einB.x);
        einB.print(30);
        einB.print(2.5);

        C einC = new C();
        einC.print(40);
        einC.print(2.5);
        System.out.println("C: " + einC.x);
    }
}
```

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

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

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

    public void print(double d)
    {
        System.out.println("A: double");
    }
}
```

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

    public void print(double d)
    {
        System.out.println("B: double");
    }
}
```

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

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

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

Ausgaben: