

Probeklausur – Java

Einführung in die Informatik

Wintersemester 2016/2017

Hinweis: Diese Probeklausur ist eine kleine Aufgabensammlung, die etwa dem Schwierigkeitsgrad der schriftlichen Prüfung des Moduls Einführung in die Informatik entspricht. Die hier zur Verfügung gestellten Aufgaben decken jedoch nicht alle behandelten Themenbereiche ab. Darüber hinaus wird es in der Klausur Wissensaufgaben geben, welche aus allen Themen und sämtlichen Teilen der Veranstaltung (Vorlesung, Tutorium und Hausaufgaben) kommen können. In den Klausuren für dieses Semester wird die Gewichtung ca. 60 % Programmieren und 40 % Rechneraufbau sein.

Aufgabe 1 (Allgemeine Fragen (Java)).

In dieser Aufgabe ist jeweils genau eine Antwort richtig, welche Sie ankreuzen sollen. Kreuzen Sie pro Teilaufgabe nur ein Kästchen an. Eine richtige Antwort ergibt einen Punkt, eine falsche 0 Punkte. Es gibt keine Minuspunkte. Um ein versehentlich gesetztes Kreuz wieder zu löschen, füllen Sie das jeweilige Kästchen aus und zeichnen ein leeres daneben. Ein Beispiel:

<i>Fragetext</i>	<u>Korrektur</u>	<i>Fragetext</i>
<input checked="" type="checkbox"/> Antwort 1	→	<input type="checkbox"/> <input checked="" type="checkbox"/> Antwort 1
<input type="checkbox"/> Antwort 2		<input type="checkbox"/> Antwort 2
<input type="checkbox"/> Antwort 3		<input checked="" type="checkbox"/> Antwort 3
<input type="checkbox"/> Antwort 4		<input type="checkbox"/> Antwort 4

1. Welcher der folgenden Datentypen ist **kein** primitiver Datentyp?

- int
- double
- String
- byte

2. Welche der folgenden Anweisungen gibt einen Compilerfehler?

- `int[] d = {0,1,2,3,4,5,6,7};`
- `int[] c = {};`
- `double[] b = new int[2];`
- `String[] a;`

3. Wie oft wird die folgende Schleife durchlaufen?

```
int a = 10;
boolean run = true;
while(run){
    if(a%7 == 0){
        run = !run;
    }
    a--;
}
```

- 0
- 3
- 4
- 5

4. Welcher der folgenden Ausdrücke wird garantiert zu **true** ausgewertet?

- `(a < b) && (b < c)`
- `!(b || a)`
- `!!(a || !a)`
- `(a >= b && b >= a)`

5. Welche Aussage zu objektorientierter Programmierung in Java ist **falsch**?

- Java erlaubt es, Konstruktoren mit dem Sichtbarkeitslevel **private** zu versehen.
- Java erlaubt das Erben von mehreren Superklassen.
- Von einer Klasse dürfen im allgemeinen beliebig viele andere Klassen erben.
- Eine Klasse abstrakte Klasse kann statische Methoden enthalten.

6. Welche Aussage bezüglich des zweidimensionalen Arrays `int[] [] a = {{1,2,3},{4,5,6}}` ist korrekt?
- `a[0].length == 2` ergibt `true`
 - `a.length == 2` ergibt `false`
 - `a == b` ergibt `false` für `int[] [] b = {{1,2,3},{4,5,6}}`;
 - `int b = a[a.length]`; gibt einen Compilerfehler
7. Welcher der folgenden Datentypen ist kein primitiver Datentyp?
- `byte`
 - `String`
 - `boolean`
 - `double`
8. Welche Aussage über Konstruktoren in Java ist falsch?
- Konstruktoren haben immer den gleichen Name wie die Klasse
 - Der Konstruktor einer Superklasse kann in einer Subklasse durch `super()` aufgerufen werden
 - Es können mehrere Konstruktoren in einer Klasse definiert werden
 - In jeder Klasse muss ein Konstruktor explizit definiert sein
9. Wir haben die folgende Beziehung gegeben `public class Motorrad extends Fahrzeug{...}`?
Was entspricht dem Prinzip der Polymorphie?
- `Fahrzeug f = new Fahrzeug();`
 - `Motorrad moto = new Object();`
 - `Fahrzeug f = new Motorrad();`
 - `Motorrad moto = new Fahrzeug();`
10. Welche Aussagen über Interfaces ist falsch ?
- Die Sichtbarkeit der Methoden des Interfaces entspricht immer der des Interfaces
 - Eine Klasse, die ein Interface implementiert, muss alle Methoden des Interfaces implementieren, sodass Objekte der Klasse erzeugt werden können
 - Klassen können jeweils nur ein Interface implementieren
 - Ein Interface kann als Datentyp verwendet werden

Aufgabe 2 (Bedingte Anweisungen).

1. Betrachten Sie die folgende Java-Methode

```
boolean f(boolean x, boolean y) {  
    if(x) {  
        return y;  
    } else if(y) {  
        return x;  
    } else {  
        return x;  
    }  
}
```

(a) Stellen Sie eine Tabelle auf, die für alle möglichen Wertepaare (x, y) der Parameter den zugehörigen Rückgabewert der Methode `f()` angibt.

(b) Implementieren Sie eine möglichst einfache Methode `boolean f2(boolean x, boolean y)`, die ohne `if`-Anweisung auskommt und für alle möglichen Wertekombinationen von `x` und `y` die gleichen Werte wie die Methode `f()` zurückliefert.

2. Betrachten Sie das folgende Punktesystem zur Notenvergabe:

Punkte	Note
36-40	1
31-35	2
26-30	3
21-25	4
0-20	5

Implementieren Sie eine Methode `int getNote(int punkte)`, die als Wert die Note der übergebenen Punkte zurückliefert. Liefern Sie den Wert `-1` zurück, falls der Parameter `punkte` keine gültige Punktezahl darstellt.

Aufgabe 3 (Schleifen).

1. Welchen Wert hat `k` nach Ausführung der folgenden Anweisungen?

```
int k = 2;
for(int i = 2; i < 7; i += 2) {
    k *= 2;
}
```

2. Wandeln Sie die folgende `for`-Anweisung in eine `while`-Anweisung um.

```
for(int i = 533; i > -12; i -= 7) {
    System.out.println(i);
}
```

-
3. Implementieren Sie eine Methode `int[] range(int[] zahlen)`, die ein `int`-Array zurückgibt, welches mit den ersten drei Einträgen von `zahlen` besetzt ist, welche Dezimalziffern enthalten. Sollte `zahlen` weniger als drei (oder keine) Dezimalziffern enthalten wird das zurückgegebene Array mit Nullen aufgefüllt.

Aufgabe 4 (Rekursion).

Betrachten Sie die rekursiv definierte Funktion für alle $n \geq 3$

$$f(3) = 1$$

$$f(4) = 2$$

$$f(n) = 2 \cdot f(n-1) + 3 \cdot f(n-2) \quad \text{für } n > 4$$

Implementieren Sie eine rekursive Java-Methode, die den Wert $f(n)$ zurückliefert. Die Variable n wird hierbei als Parameter der Methode übergeben. Die Deklaration von lokalen Variablen und Verwendung von Schleifen ist nicht erlaubt. Um welche Art von Rekursion handelt es sich? Begründen Sie Ihre Antwort.

Aufgabe 5 (Klassen und Objekte).

1. Ergänzen Sie die folgende Java-Klasse wie in den Kommentaren beschrieben.

```
public class C {  
  
    private int a;  
  
    public C(int a) {  
        // initialisiere Attribut mit uebergebenen Parameter  
    }  
  
    public void f() {  
        a *= 2;  
        System.out.println("a = " + a);  
    }  
  
    public static void main(String[] args) {  
        // Erzeuge ein Objekt der Klasse C. Dabei soll der Wert des  
        // Attributs a mit 5 initialisiert werden.  
  
        // Rufe die Methode f() auf.  
    }  
}
```

2. Gegeben ist die Klasse Fortbewegungsmittel.

```
public class Fortbewegungsmittel {  
  
    private int speed;  
    private String besitzer;  
  
    public Fortbewegungsmittel(String besitzer){  
        this.besitzer = besitzer;  
        speed = 0;  
    }  
  
    public int getSpeed(){  
        return speed;  
    }  
  
    public void beschleunigen(int deltaSpeed){  
        speed += deltaSpeed;  
    }  
}
```

Implementieren Sie eine Klasse `Fahrrad`, welche von `Fortbewegungsmittel` erbt. Die Zugriffskriterien für Konstruktoren, Methoden und Attribute sind wie in der Superklasse.

Die Klasse `Fahrrad` soll folgenden Anforderungen genügen:

- (a) Sie besitzt zwei zusätzliche Attribute: `gang` vom Typ `int` sowie `locked` vom Typ `boolean`.
- (b) Sowohl die Geschwindigkeit (`speed`) als auch der derzeitige Gang dürfen nicht negativ werden.

- (c) Die Klasse besitzt einen parametrisierten Konstruktor, der Parameter zum Setzen der Attribute **besitzer**, **gang** und **locked** übergeben bekommt. Die Geschwindigkeit soll (wie in **Fortbewegungsmittel**) als 0 initialisiert werden.
- (d) Die Klasse besitzt einen weiteren parameterlosen Konstruktor, der die Attribute **gang** und **speed** mit 0, das Attribut **locked** mit **false** und den Besitzer mit dem String "Default" initialisiert.
- (e) Um die zuvor erwähnten Eigenschaften (siehe b)) des Attributs **speed** zu erfüllen, muss die Methode **beschleunigen(...)** der Superklasse überschrieben werden.
- (f) Eine Setter-Methode für **gang** soll implementiert werden, welche den Eigenschaften des Attributs (siehe b)) gerecht wird.

Aufgabe 7 (Fließkommazahlen).

1. Addieren Sie die Zahlen 45_{10} und 0.0625_{10} im angegebenen Gleitkommaformat und konvertieren Sie das Ergebnis zurück in eine Dezimalzahl.

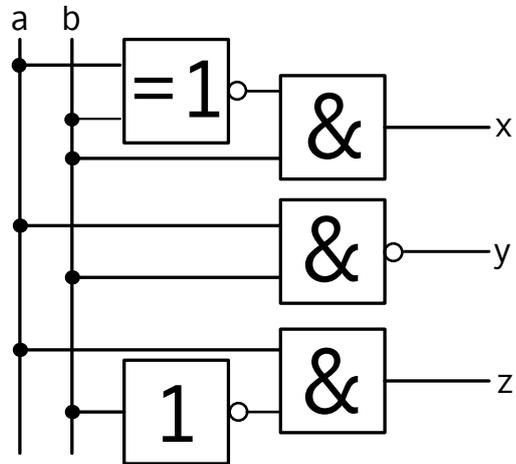
Vorzeichen	Exponent	Mantisse
1 Bit	4 Bits	5 Bits

Der Exzess beträgt: $7_{10} = 0111_2$

2. Wie groß ist der Rundungsfehler, der bei der Addition in der ersten Teilaufgabe entsteht. Geben Sie den Rundungsfehler als Dezimalzahl an.

Aufgabe 8 (Schaltungen).

1. Stellen Sie die Wertetabelle für folgende Schaltung auf:



(a und b sind die Eingänge, x, y und z sind die Ausgänge der Schaltung.)

2. Zeichnen Sie eine Schaltung, die der folgenden Wertetabelle entspricht.

a	b	c	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(*a, b und c sind die Eingänge, y ist der Ausgang der Schaltung.*)