

# Einführung in die Informatik (Java) - Probeklausur

11.02.2019 - 15.02.2019

Name: .....

Matr.-Nr.: .....

**Bearbeitungszeit: 120 Minuten**

Aufgabe	Punkte	Erreichte Punkte
<b>Java</b>		
1	10	
2	6	
3	10	
4	14	
5	8	
6	12	
<b>Rechneraufbau</b>		
7	10	
8	6	
9	7	
10	10	
11	7	
<b>Summe</b>	100	

**Hinweise:**

- Tragen Sie auf **allen** Blättern Ihren Namen und Ihre Matrikelnummer ein.
- Als Hilfsmittel sind nur Stift und Lineal zugelassen.
- Verwenden Sie für die Lösung der Aufgaben **nur** das mit diesem Deckblatt ausgeteilte Papier. Verwenden Sie auch die Rückseiten der Blätter und die Zusatzblätter am Ende der Klausur. Weitere Blätter werden auf Anfrage zur Verfügung gestellt. Lösungen auf anderem Papier werden **nicht** bewertet.
- Wenn Sie Ihre Antworten nicht an den vorgesehenen Stellen platzieren, kennzeichnen Sie dies deutlich auf der Aufgabenseite, damit Ihre Antwort bei der Bewertung berücksichtigt wird (z.B. „Antwort siehe Seite XY“).
- Schreiben Sie **nicht** mit Bleistift und **nicht** mit rotem oder grünem Stift.
- Wenn Sie in einer Aufgabe nur eine Methode implementieren sollen, brauchen Sie die umgebenden Klasse nicht mit angeben.
- Lassen Sie bei jeder Rechnung den Lösungsweg erkennen.
- Unleserliche oder mehrdeutige Lösungen werden nicht gewertet.

**Wir wünschen Ihnen viel Erfolg!**

**Aufgabe 1 (10 Punkte) 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.

1. (1 Punkt) Welcher primitive Datentyp in Java ist kein Ganzzahldatentyp?

- byte
- short
- float
- char

2. (1 Punkt) Beim Aufruf der Methode `void fun(Object a)` wird ...

- ... ein Objekt mit Namen `a` in der Methode erzeugt.
- ... eine Referenz mit dem Namen `a` auf das übergebene Objekt erzeugt.
- ... das Objekt `a` kopiert.
- ... der Zugriff auf ein äußeres Objekt `a` ermöglicht.

3. (1 Punkt) Welche der folgenden Aussagen ist falsch?

Verwendet man eine Superklasse als Datentyp für Variablen ...

- ... koennen diese Objekte verschiedener Subklassen aufnehmen.
- ... koennen Methoden die auf Subklassenebene eingeführt werden nur durch Casting aufgerufen werden.
- ... kann man keine Objekte von Subklassen aufnehmen. Dies geht nur anders herum.
- ... kann es sinnvoll sein die `instanceof`-Operation zu verwenden.

4. (1 Punkt) Welches ist kein Schlüsselwort in Java?

- `protected`
- `abstract`
- `toString`
- `instaceof`

5. (1 Punkt) Es wurde ein `double`-Array `arr` mit 4 Einträgen angelegt. Welche Anweisung ist fehlerhaft?

- `arr[4-arr.length] = 2.3;`
- `arr[(int)3.2] = 0.1;`
- `arr[arr.length] = 1.4;`
- `arr[7-4] = 2.0;`

6. (1 Punkt) Welcher der folgenden Ausdrücke evaluiert zu `true`, wenn die Variablen die Werte `a = 2`; `b = true`; `c = false` haben?

- `(a < 2 || b && c )`
- `(a == 2)&& (b != c)`
- `((a >= 2)&& !b)&& c`
- `(a>2)|| (b != true)|| c`

7. (1 Punkt) Wenn man in einer Subklasse eine Methode mit dem selben Namen wie in der Superklasse jedoch mit anderer Anzahl von Parametern implementiert, spricht man von:

- Verschatten
- Überschreiben
- Überladen
- Verbergen

8. (1 Punkt) Welche der folgenden Zeilen würde einen Fehler verursachen?

- `int[] i;`
- `double[] d={};`
- `float[] f={5,500,50000,50000000};`
- `byte[] b={1.2,22.0,6.0};`

9. (1 Punkt) Welche Aussage bezüglich Interfaces in Java ist falsch?

- Interfaces können als Datentyp verwendet werden.
- Interfaces benötigen das Schlüsselwort **abstract**.
- Interfaces beinhalten nur abstrakte Methoden.
- Methodendeklarationen in Interfaces benötigen keine Sichtbarkeitsspezifikation

10. (1 Punkt) Gegeben sei die Objektklasse Buch. Wie viele Referenzen werden in folgender Zeile erzeugt:

```
Buch[] bibliothek =new Buch[30];
```

- 1
- 30
- keine
- 31

**Aufgabe 2 (6 Punkte) Bedingte Programmausführung (Java).**

Gehen Sie davon aus Sie haben eine Gemäldesammlung und möchten nun zu jedem Bild den Wert des Bildes zuordnen. Der Wert bestimmt sich folgendermaßen:

- Große Bilder ( $> 1.2$  qm) von Picasso mit Echtheitszertifikat: 10000000.00 Euro
- Kleinere Bilder von Picasso mit Echtheitszertifikat: 3000000.00 Euro
- Sonstige sehr große Bilder ( $> 2.5$  qm): 2000.00 Euro
- Sonstige nicht sehr große Bilder: 120.99 Euro

Implementieren Sie dazu eine Methode `wertermittlung`, welche den Wert in Euro ermittelt und zurückgibt. Die Methode bekommt drei Übergabeparameter:

- `flaeche` (die Fläche des Bildes in Quadratmetern),
- `ispicasso` (von Picasso gemalt),
- `zertifiziert` (Echtheit ist zertifiziert).

Überlegen Sie sich sinnvolle Datentypen für den Rückgabewert und die Parameter. Gehen Sie davon aus, dass die Fläche des Bildes immer positiv ist.

**Aufgabe 3 (10 Punkte) Schleifen (Java).**

1. (4 Punkte) Im folgenden sollen Sie die vorgegebene `while`-Schleife in eine `for`-Schleife umwandeln. Schreiben Sie dazu die gesamte Methode.

```
public static int geradeSumme(int[] array){
    int summeDerGeradenZahlen = 0;
    int j = 0;
    while(j<array.length){
        j++;
        if(array[j-1]%2 == 0){
            summeDerGeradenZahlen += array[j-1];
        }
    }
    return summeDerGeradenZahlen;
}
```

2. (6 Punkte)

Entwerfen Sie eine statische Methode `decodiere`. Ihr werden zwei Arrays übergeben (`int [] indices` und `char [] buchstaben`) und sie gibt einen `String` zurück. Die Werte in `indices` zeigen an welches Element aus `buchstaben` an die Position im `String` geschrieben wird (siehe Beispiel). Durchlaufen Sie dazu das `indices`-Array von Anfang bis Ende und hängen Sie den passenden Buchstaben an den `String` an. Sie können davon ausgehen, dass alle Werte in `indices` zwischen 0 und `buchstaben.length-1` liegen.

Beispiel: Für `indices = {0,1,2,2,1,2,2,1,3,3,1}` und `buchstaben = {'M', 'i', 's', 'p'}` ist der `String` *"Mississippi"*

**Aufgabe 4 (14 Punkte) Klassen (Java).**

1. (5 Punkte) Implementieren Sie eine Klasse **Fahrzeug**. Stellen Sie bei der Implementierung dieser Aufgabe sicher, dass:

- auf die Klasse, den Konstruktor und die Methoden von jeder anderen Klasse aus zugegriffen werden kann und
- ein direkter Zugriff auf die Attribute von außen nicht möglich ist.

Die Klasse **Fahrzeug** besitzt folgende Attribute:

- Ein Attribut **name** vom Typ **String**, das den Namen bzw. den Fahrzeugtyp angibt
- Ein Attribut **geschw** vom Typ **double**, das die Geschwindigkeit des Fahrzeugs angibt

Implementieren Sie zudem folgende Methoden:

- Einen parametrisierten Konstruktor, der **name** mit dem übergebenen Parameter initialisiert und die Geschwindigkeit auf 0 setzt.
- Eine Methode **void beschleunigen(double x)**, die die Geschwindigkeit um **x** erhöht.
- Getter für beide Attribute

2. (6 Punkte) Implementieren Sie eine Klasse `PKW` die von `Fahrzeug` erbt und die gleichen Zugriffskriterien wie `Fahrzeug` erfüllt. Die Klasse `PKW` besitzt ein Attribut `hoechstgeschw` vom Typ `double` das die Höchstgeschwindigkeit angibt. Implementieren sie außerdem folgende Methoden:

- Einen parametrisierten Konstruktor, der `name` und `hoechstgeschw` mit dem übergebenen Parametern initialisieren soll.
- Eine Methode `void beschleunigen(double x)`, die die Methode aus der Superklasse überschreibt und mit Hilfe von `hoechstgeschw` überprüft, ob eine Beschleunigung möglich ist und soweit es die `hoechstgeschw` zulässt, umsetzt.

3. (3 Punkte) Ergänzen Sie die folgende Testklasse, indem Sie jeweils ein Objekt von `PKW` und `Fahrzeug` erzeugen. Rufen Sie die `beschleunigen`-Methode für eines der Objekte auf.

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

**Aufgabe 5 (8 Punkte) Rekursion (Java).**

1. (3 Punkte) Betrachten Sie den Algorithmus für das Generieren von Dualzahlen aus Dezimalzahlen. Wandeln Sie dazu schrittweise  $5_{(10)}$  in eine Dualzahl um. Markieren Sie die Rekursionsschritte. Markieren Sie weiter, welche Zahl an den nächsten Rekursionsschritt übergeben und welche an die Dualzahl angehängt wird. Markieren Sie den Grundzustand, bei dem die Rekursion nicht weiter läuft.

2. (5 Punkte) Schreiben Sie nun eine statische, rekursive Methode `String todual(int x)`, die eine natürliche Zahl übergeben bekommt und diese in eine Binärzahl umwandelt und als String zurückgibt.

*Hinweis:* Modulo liefert den Rest einer Ganzzahldivision.





4. (4 Punkte) Implementieren Sie eine Testklasse `TestInkrement` mit der Einstiegsmethode für die Ausführung. Legen Sie in dieser ein Array der Länge 3 an, welches ein Objekt vom Typ `Uhrzeit` und zwei Objekte vom Typ `Ganzzahl` aufnimmt. Bei der Objekterzeugung benutzen sie sowohl den parameterlosen und den paramtrischen Konstruktor (initialisiert `zahl` mit 15) der Klasse `Ganzzahl`. Wenden Sie nun für jedes Objekt im Array die Methode `bildeInkrement` an.

**Aufgabe 7 (10 Punkte) Allgemeine Fragen (Rechneraufbau).**

In dieser Aufgabe ist jeweils genau eine Antwort richtig, welche Sie ankreuzen sollten. 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.

1. (1 Punkt) Welche Aussage über den Prozessorbus trifft zu:
  - Der Adressbus schreibt und liest Adressen in den Arbeitsspeicher.
  - Der Steuerbus enthält Interruptsleitungen
  - Ein Bussystem mit höherer Taktrate hat grundsätzlich mehr Bandbreite.
  - Der Grafikbus steuert den Bildschirm an.
  
2. (1 Punkt) Welche Aussage zu Bereichsüberschreitungen in Ganzzahldarstellungen ist korrekt?
  - Bei Subtraktionen benötigt man mehr als ein Checkbit um sicherzustellen, dass kein Überlauf stattfindet.
  - Bereichsüberschreitungen treten nur bei der Addition bzw. Subtraktion mit Checkbits auf.
  - Die Addition zweier positiver Zahlen kann ein negatives Ergebnis liefern.
  - Die Zweierkomplementsdarstellung verhindert Bereichsüberschreitungen.
  
3. (1 Punkt) Welche Aussage zu Voll- bzw. Halbaddierern ist **falsch**?
  - Der mögliche Wertebereich eines Volladdierers ist doppelt so groß wie der eines Halbaddierers.
  - Ein Volladdierer besteht aus 2 Halbaddierern und einem OR-Gatter.
  - Je ein XOR- und AND-Gatter bilden einen Halbaddierer.
  - Beide besitzen die gleiche Anzahl von Ausgängen.
  
4. (1 Punkt) Welche aufsteigende Reihenfolge von Speichern ist korrekt, wenn mit dem kleinsten/schnellsten Speicher begonnen wird?
  - 1. Register, 2. RAM, 3. Cache, 4. Festplatten
  - 1. Festplatten, 2. RAM, 3. Register, 4. Cache
  - 1. Register, 2. Cache, 3. RAM, 4. Festplatten
  - 1. Cache, 2. Register, 3. RAM, 4. Festplatten
  
5. (1 Punkt) Was ist **keine** Funktion des Betriebssystem-Kerns:
  - Bereitstellen der Eingabeaufforderung.
  - Kontrolle der Hardware.
  - Bereitstellen von Systemfunktionen.
  - Datei & Prozessverwaltung.
  
6. (1 Punkt) Der Wertebereich von 4-Bit Dualzahlen mit einem Exzess von 2 ist:
  - 2 ... 13
  - 0 ... 15
  - 2 ... 17
  - 7 ... 8

7. (1 Punkt) Welche Aussage zur ALU stimmt **nicht**?

- Sie dient zur Durchführung von Rechenoperationen (z.B. Addition/Subtraktion).
- Sie kann booleschen Operationen (z.B. XOR) auswerten.
- Ist Teil eines Schaltwerk.
- Sie ist im Steuerwerk untergebracht

8. (1 Punkt) Ein beendeter noch existierender Prozess wird auch bezeichnet als:

- Leiche
- Golem
- Zombie
- Vampir

9. (1 Punkt) Welches Gesetz bzw. welche Regel gilt **nicht** für die boolesche Algebra?

- Komplement
- Idempotenz
- Absorption
- Induktion

10. (1 Punkt) Welches Bauelement eignet sich **nicht** zum Speichern?

- Multiplexer
- Flip-Flop
- Binary Cell
- Register

**Aufgabe 8 (6 Punkte) Zahlendarstellung (Rechneraufbau).**

1. (2 Punkte) Wandeln Sie die Zahl  $48_{(15)}$  (zur Basis 15) in eine Zahl zur Basis 9 um. Der Lösungsweg muss erkennbar sein.

2. (2 Punkte) Wandeln Sie die Dezimalzahl  $45_{(10)}$  in eine Hexadezimalzahl um. Der Lösungsweg muss erkennbar sein.

3. (2 Punkte) Wandeln Sie die Dualzahl  $101110,011_{(2)}$  in eine Oktalzahl um. Der Lösungsweg muss erkennbar sein.



**Aufgabe 10 (10 Punkte) Gleitkommadarstellung (Rechneraufbau).**

1. (4 Punkte) Wandeln Sie folgende rationale Zahl  $-5.45_{(10)}$  in die Gleitkommadarstellung um. Der Lösungsweg muss dabei detailliert erkennbar sein.

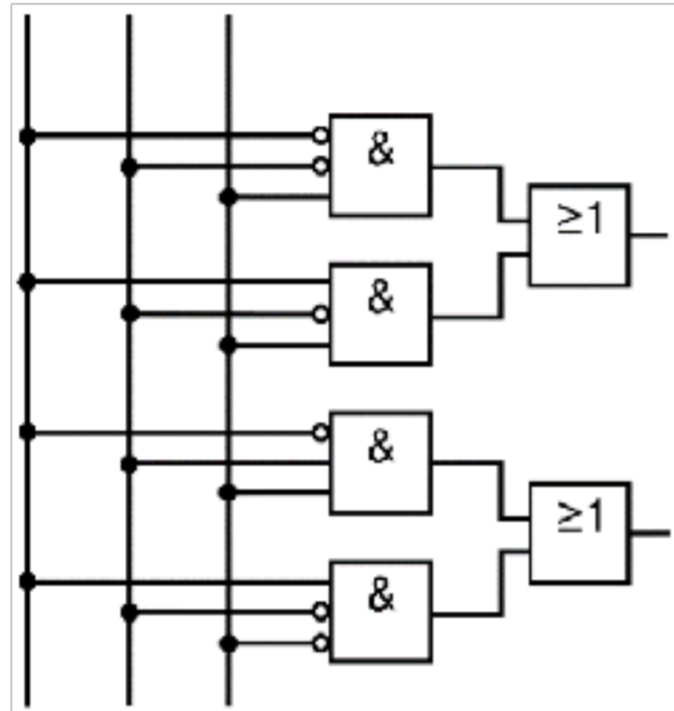
*Hinweis:* Es soll folgendes Darstellungsformat angenommen werden: (1Bit Vorzeichen, 4 Bit Exponent, 5 Bit Mantisse, Exzess = 7)

2. (6 Punkte) Addieren Sie die Gleitkommazahlen 0 0101 10110 und 0 0011 01001. Wie groß ist der dort entstandene Rundungsfehler? Der Rundungsfehler entspricht der Wertigkeit der wegfallenden Stellen und kann in Form einer Zweierpotenz angegeben werden.

*Hinweis:* Es soll folgendes Darstellungsformat angenommen werden: (1Bit Vorzeichen, 4 Bit Exponent, 5 Bit Mantisse, Exzess = 7)

**Aufgabe 11 (7 Punkte) Schaltungen & Boolesche Ausdrücke (Rechneraufbau).**

1. (3 Punkte) Beschriften Sie die Ein- und Ausgänge der nachfolgenden Schaltung, so dass die Tabelle erfüllt ist.



$x_0$	$x_1$	$x_2$	$y_0$	$y_1$
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	0	0

2. (4 Punkte) Überprüfen Sie mittels einer Wertetabelle ob der gegebene boolesche Ausdruck zu  $y_0$  oder  $y_1$  äquivalent ist.

$$(x_2 \oplus x_0) (\bar{x}_0 \bar{x}_1 + x_0 x_1)$$

$x_0$	$x_1$	$x_2$	





