

# Einführung in die Informatik (Java) - Probeklausur

12.–18.02.2024

## Musterlösung Stand: 19. Juli 2023

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	

**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`.
  - Die Methodendeklarationen in einem Interfaces sind implizit abstrakt.
  - Methodendeklarationen in Interfaces benötigen keine Sichtbarkeitspezifikation
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.

### Lösung:

```

/*
Ein Kunsthändler will den Verkaufspreis seiner Bilder automatisiert ermitteln.
Implementieren
Sie dazu eine öffentliche Methode preisermittlung, welche den Verkaufspreis in Euro
ermittelt und
zurückgibt.
Die Methode bekommt drei Übergabeparameter:
flaeche (die Fläche des Bildes in Quadratmetern),
picasso (die Information ob das Bild von Picasso gemalt wurde),
zertifikat (die Information ob die Echtheit des Bildes von einem Experten bestätigt
wurde).
Überlegen Sie sich die sinnvollsten Rückgabe- als auch Übergabeparameter für die Methode
preisermittlung. Gehen Sie davon aus, dass die Fläche des Bilds immer positiv ist.

- Bilder mit mindestens 1,2qm Fläche von Picasso mit Zertifikat: 10000000.00Euro
- kleinere Bilder von Picasso mit Zertifikat: 3000000.00Euro
- sonstige Bilder mit mindestens 2,5qm Fläche: 2000.00Euro
- kleinere sonstige Bilder: 120,99Euro
*/

/*Nur damit ausführbar, nicht Teil der Aufgabe*/
public class IFAufgabe{
    public static void main (String[] args){
        System.out.println(preisermittlung(true, false, 2));
    }
}
/*Nur damit ausführbar, nicht Teil der Aufgabe*/

public static double preisermittlung(boolean picasso, boolean zertifikat, double
flaeche){
    if(picasso && zertifikat){
        if(flaeche >= 1.2){
            return 10000000.00;
        }
    }
    else{
        return 3000000.00;
    }
}

```

```
    }  
  }  
  else{  
    if(flaeche >= 2.5){  
      return 2000.00;  
    }  
    else{  
      return 120.99;  
    }  
  }  
}  
  
/*Nur damit ausführbar, nicht Teil der Aufgabe*/  
}  
/*Nur damit ausführbar, nicht Teil der Aufgabe*/
```

## 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;
}
```

**Lösung:**

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

**Lösung:**

```
public static String decodiere(int[] indices, char[] buchstaben) {
    String str = "";
    for (int i=0;i<indices.length;i++) {
        str += buchstaben[indices[i]];
    }
    return str;
}
```

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

- (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

**Lösung:**

```
class Fahrzeug{

    private String name;
    private double geschw;

    public Fahrzeug(String name){
        this.name=name;
        geschw=0;
    }

    public void beschleunigen(double x){
        geschw+=x;
    }

    public void setName(String name){
        this.name=name;
    }

    public String getName(){
        return name;
    }

    public void setGeschw(double geschw){
        this.geschw=geschw;
    }

    public double getGeschw(){
        return geschw;
    }

}
```

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 uns soweit es die `hoechstgeschw` zulaesst, umsetzt.

**Lösung:**

```
class PKW extends Fahrzeug{

    private double hoechstgeschw;

    public PKW(String name, double hoechstgeschw){
        super(name);
        this.hoechstgeschw=hoechstgeschw;
    }

    public void beschleunigen(double x){
        if(super.getGeschw()<hoechstgeschw){
            if(getGeschw()+x>hoechstgeschw)
                super.beschleunigen(hoechstgeschw-getGeschw());
            else
                super.beschleunigen(x);
        }
    }
}
```

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) {

    }
}
```

**Lösung:**

```
public class Fuhrpark {
    public static void main(String[] args) {
        PKW auto = new PKW("E-Auto", 250);
        Fahrzeug f = new Fahrzeug("Boot");
        auto.beschleunigen(25);
    }
}
```



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

- (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.

**Lösung:**

5 / 2 → 2 R 1 .... 2 fuer den naechsten Rekursionsschritt und 1 fuer String  
2 / 2 → 1 R 0 .... 1 fuer den naechsten Rekursionsschritt und 0 fuer String  
1 / 2 → 0 R 1 .... 0 → Grundzustand erreicht und 1 fuer den String

- (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.

**Lösung:**

```
public class BinRek {  
  
    /**  
     * Aufgabe: Schreiben Sie eine statische, rekursive Methode, die eine nat"urliche  
     * Zahl "ubergeben  
     * bekommt und diese in eine Bin"arzahl umwandelt und zur"uckgibt.  
     * Die R"uckgabe der Bin"arzahl soll dabei als String erfolgen.  
     */  
    public static void main(String[] args) {  
        int a = 121;  
        System.out.println(a + " als Binaerzahl: " + dual(a));  
    }  
  
    public static String dual(int dez){  
        if(dez == 0) return "";  
        else{  
            return dual(dez/2)+dez%2;  
        }  
    }  
}
```

## Aufgabe 6 (12 Punkte) Interfaces (Java).

1. (2 Punkte) Implementieren Sie ein öffentliches Interface **Inkrementierbar**, welches die nach außen sichtbare parameterlose Methode `bildeInkrement` deklariert. Der Rückgabebetyp der Methode ist `void`.

**Lösung:**

```
public interface Inkrementierbar{
    public void bildeInkrement();
}
```

2. (3 Punkte) Implementieren Sie weiterhin die öffentliche Klasse **Uhrzeit**, welche das Interface **Inkrementierbar** implementiert. Sie enthält die privaten Attribute `minute` und `stunde`. `bildeInkrement` erhöht `minute` um 1, falls `minute` kleiner als 60 ist. Ansonsten wird `minute` auf 0 gesetzt und `stunde` um 1 erhöht.

**Lösung:**

```
public class Uhrzeit implements Inkrementierbar{
    private int minute;
    private int stunde;

    public void bildeInkrement(){
        if(minute<60){
            minute++;
        }else{
            stunde++;
            minute=0;
        }
    }
}
```

3. (3 Punkte) Implementieren Sie weiter die öffentliche Klasse **Ganzzahl**, welche das Interface **Inkrementierbar** implementiert. Sie enthält das private Attribut `zahl`. Zusätzlich soll **Ganzzahl** einen unparametrisierten und einen parametrisierten Konstruktor enthalten. `bildeInkrement` erhöht `zahl` um 1.

**Lösung:**

```
public class Ganzzahl implements Inkrementierbar{
    private int zahl;

    public Ganzzahl(){
        zahl = 0;
    }

    public Ganzzahl(int zahl){
        this.zahl = zahl;
    }

    public void bildeInkrement(){
        zahl++;
    }
}
```

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.

**Lösung:**

```
public class TestInkrement{
    public static void main(String[] args){
        Inkrementierbar test[] = new Inkrementierbar[3];
        test[0] = new Ganzzahl();
        test[1] = new Ganzzahl(-7);
        test[2] = new Uhrzeit();

        // erste Loesungsmoeglichkeit
        for(int i=0;i<test.length;i++){
            test[i].bildeInkrement();
        }

        // weitere Loesungsmoeglichkeit
        test[0].bildeInkrement();
        test[1].bildeInkrement();
        test[2].bildeInkrement();
    }
}
```

**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.

**Lösung:**

$$48_{(15)} = 4 \cdot 15^1 + 8 \cdot 15^0 = 60 + 8 = 68_{(10)}$$

$$68 : 9 = 7 \text{ Rest } 5 \quad (z_0 = 5)$$

$$7 : 9 = 0 \text{ Rest } 7 \quad (z_1 = 7)$$

$$\text{Also ist } 48_{15} = 75_9.$$

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

**Lösung:**

$$45 : 16 = 2 \text{ Rest } 13 \quad (z_0 = D)$$

$$2 : 16 = 0 \text{ Rest } 2 \quad (z_1 = 2)$$

$$\text{Also ist } 45_{10} = 2D_{16}.$$

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

**Lösung:**

$$101110,011_{(2)} = 101|110,011 = 56,3_{(8)}$$

**Aufgabe 9 (7 Punkte) Ganzzahldarstellung (Rechneraufbau).**

1. (2 Punkte) Wandeln Sie die Dezimalzahl **-13** in eine Zweierkomplementdarstellung mit 6-Bit um. Alle Schritte des Lösungswegs müssen detailliert dargestellt werden.

**Lösung:**

$-13_{(10)} \rightarrow 110011$   
 $13_{(10)}$  ins Dualsystem 001101  
 001101 invertieren  $\rightarrow 110010$   
 1 addieren  $\rightarrow 110011$

2. (1 Punkt) Wandeln Sie die Zahl in 6-Bit-Zweierkomplementdarstellung (**011011**) in eine Dezimalzahl um. Alle Schritte des Lösungswegs müssen detailliert dargestellt werden.

**Lösung:**

$011011 \rightarrow 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 16 + 8 + 2 + 1 = 27$

3. (4 Punkte) Führen Sie die nachfolgenden Operation mit Hilfe des 6-Bit Zweierkomplements aus. Testen Sie auf Überlauf/Unterlauf und geben Sie an um welchen es sich handelt, sollte es zu einem kommen. Wandeln Sie das Ergebnis in jedem Fall wieder in Dezimaldarstellung um. Der Lösungsweg muss detailliert dargestellt werden:

$$-13_{(10)} + 27_{(10)}$$

**Lösung:**

Binär:	110011
	011011
+ Checkbit	[1]110011
	[0]011011
Übertr.:	1[1]100110
Summe	1[0]001110
Check:	ok

Kein Unter/Überlauf, da das Checkbit mit dem Vorzeichenbit des Ergebnisses übereinstimmt: [0]0. Das Ergebnis 14 passt also in eine 6-Bit Zweierkomplement Zahl.

$001110 \rightarrow 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 = 8 + 4 + 2 = 14$

$011011 \rightarrow 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 \rightarrow 16 + 8 + 2 + 1 = 27$

**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)

**Lösung:**

$$-5 \rightarrow -101$$

$$0,45 \rightarrow 0.01110011$$

$$0,45 * 2 = 0,9 (z_{-1} = 0)$$

$$0,9 * 2 = 1,8 (z_{-2} = 1)$$

$$0,8 * 2 = 1,6 (z_{-3} = 1)$$

$$0,6 * 2 = 1,2 (z_{-4} = 1)$$

$$0,2 * 2 = 0,4 (z_{-5} = 0)$$

$$0,4 * 2 = 0,8 (z_{-6} = 0)$$

$$-5.45 \rightarrow -101.011100$$

$$-1.01011100 * 2^2$$

$$\text{Vorzeichen} = 1$$

$$\text{Exponent} = 2 + 7 = 9 \rightarrow 1001$$

$$\text{Verk. Mantisse} = .01011$$

$$\Rightarrow 1100101011$$

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)

**Lösung:**

$$1.10110 \cdot 2^{5-7} + 1.01001 \cdot 2^{3-7}$$

$$\text{Exponent anpassen: } 1.10110 \cdot 2^{5-7} + 0.0101001 \cdot 2^{5-7}$$

$$\text{Genauigkeitsverlust: } 0.0000001 \cdot 2^{-2} \rightarrow 2^{-9}$$

Addieren:

1.10110	00	⇒	1.0 · 2 <sup>6-7</sup>
+0.01010	01		
10.00000	01		

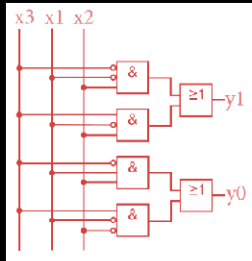
$$\text{Vorzeichen: } 0$$

$$\text{Exponent: } 0110$$

$$\text{Verk. Mantisse: } 00000$$

$$\text{Ergebnis} \Rightarrow 0011000000$$





x1	x2	x3	$(x3 \oplus x1) * ((\overline{x1} * \overline{x2}) + (x1 * x2))$				
0	0	0	0	0	1	1	0
0	0	1	1	1	1	1	0
0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	1	0	1	1
1	1	1	0	0	0	1	1

↑

=> Ist Äquivalent zu y0 !



