

# **Folge 3.3**

## **for-Schleifen**

**Und viele Aufgaben zu Schleifen**

## 3.3 for-Schleifen

### Beispiel 1

```
int sum = 0;  
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

## 3.3 for-Schleifen

### Beispiel 1

```
int sum = 0;  
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

```
sum = 0  
sum = 0 + 1 = 1  
sum = 1 + 2 = 3  
sum = 3 + 3 = 6  
sum = 6 + 4 = 10  
sum = 10 + 5 = 15  
sum = 15 + 6 = 21  
sum = 21 + 7 = 28  
sum = 28 + 8 = 36  
sum = 36 + 9 = 45  
sum = 45 + 10 = 55
```

## 3.3 for-Schleifen

### Beispiel 2

```
int sum = 0;  
for (int i = 5; i <= 8; i++)  
    sum = sum + i*i;
```

```
sum = 0  
sum = 0 + 5*5 = 25  
sum = 25 + 6*6 = 61  
sum = 61 + 7*7 = 110  
sum = 110 + 8*8 = 174
```

## 3.3 for-Schleifen

### Beispiel

*Laufvariable*



```
graph TD; A[Laufvariable] --> B[for (int i = 1; i <= 10; i++)  
sum = sum + i;]; A --> C[for (int i = 5; i <= 8; i++)  
sum = sum + i*i;];
```

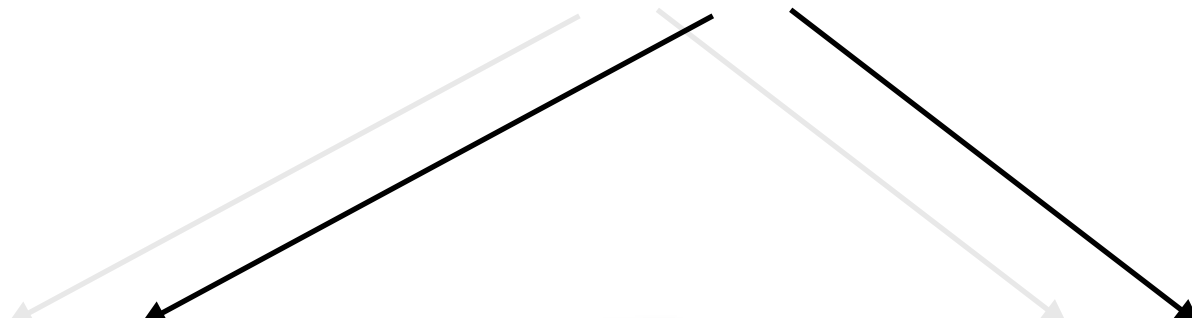
```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

```
for (int i = 5; i <= 8; i++)  
    sum = sum + i*i;
```

## 3.3 for-Schleifen

### Beispiel

Startwert



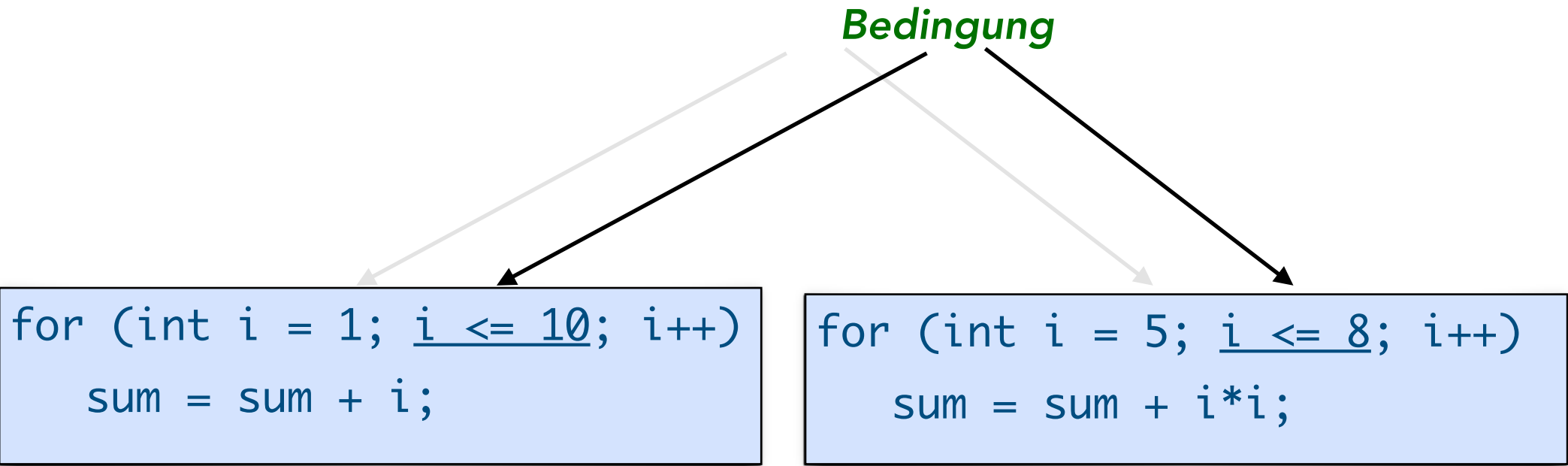
```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

```
for (int i = 5; i <= 8; i++)  
    sum = sum + i*i;
```

## 3.3 for-Schleifen

### Beispiel

*Bedingung*




```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

```
for (int i = 5; i <= 8; i++)  
    sum = sum + i*i;
```

## 3.3 for-Schleifen

### Beispiel

*Inkrement*



```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

```
for (int i = 5; i <= 8; i++)  
    sum = sum + i*i;
```



## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)
```

```
for (int k = 5; k <= 25; k++)
```

```
for (int mon = 12; mon >= 1; mon--)
```

```
for (double d = 3.1; d > 0; d -= 0.1)
```

## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)
```

```
for (int k = 5; k <= 25; k++)
```

Die Laufvariable muss  
nicht bei 1 starten!



```
for (int mon = 12; mon >= 1; mon--)
```

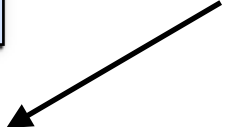
```
for (double d = 3.1; d > 0; d -= 0.1)
```

## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)
```

Die Laufvariable muss nicht bei 1 starten!



```
for (int k = 5; k <= 25; k++)
```

Die Laufvariable kann auch dekrementiert werden



```
for (int mon = 12; mon >= 1; mon--)
```

```
for (double d = 3.1; d > 0; d -= 0.1)
```

## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)
```

Die Laufvariable muss nicht bei 1 starten.



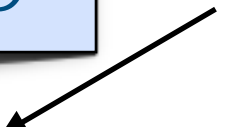
```
for (int k = 5; k <= 25; k++)
```

Die Laufvariable kann auch dekrementiert werden.



```
for (int mon = 12; mon >= 1; mon--)
```

Die Laufvariable kann auch vom Typ double sein.



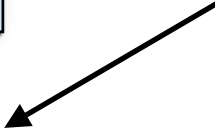
```
for (double d = 3.1; d > 0; d -= 0.1)
```

## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

Die Laufvariable kann auch vor der for-Schleife deklariert werden.



```
int i;  
for (i = 1; i <= 10; i++)  
    sum = sum + i;
```

## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;  
int y = i;
```

**Fehlermeldung!**

Fehlermeldung!  
Variable i ist unbekannt!

keine Fehlermeldung

```
int i;  
for (i = 1; i <= 10; i++)  
    sum = sum + i;  
int y = 1
```

## 3.3 for-Schleifen

### Die Laufvariable

```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;  
int y = i;
```

**Fehlermeldung!**

Laufvariablen, die im Schleifenkopf deklariert werden, sind nur im Schleifenkörper existent!

## 3.3 for-Schleifen

### Die Schleifenbedingung

```
for (int i = 1; i <= 10; i++)  
    sum = sum + i;
```

Solange die Schleifenbedingung den Wert true liefert, wird der Schleifenkörper ausgeführt.



## 3.3 for-Schleifen

### Die Schleifenbedingung

```
for (int i = 1; i > 10; i++)  
    sum = sum + i;
```

## 3.3 for-Schleifen

### Die Schleifenbedingung

```
for (int i = 1; i > 10; i++)  
    sum = sum + i;
```

**Fehlerhafte Schleifenbedingung!**  
Warum?

## 3.3 for-Schleifen

### Die Schleifenbedingung

```
for (int i = 1; i > 10; i++)  
    sum = sum + i;
```

#### Fehlerhafte Schleifenbedingung!

Warum?

- i hat den Wert 1
- Die Bedingung  $i > 10$  ist bei Beginn der Schleife nicht erfüllt.
- Die Schleife wird nicht ausgeführt!

## 3.3 for-Schleifen

### Die Schleifenbedingung

```
for (int i = 1; i > 0; i++)  
    sum = sum + i;
```

**Fehlerhafte Schleifenbedingung!**  
Warum?

## 3.3 for-Schleifen

### Die Schleifenbedingung

```
for (int i = 1; i > 0; i++)  
    sum = sum + i;
```

#### Fehlerhafte Schleifenbedingung!

Warum?

- i hat den Wert 1
- Die Bedingung  $i > 0$  ist bei Beginn der Schleife erfüllt.
- Mit jeder Inkrementation wird i noch größer, die Bedingung bleibt true.
- Das führt zu einer **Endlosschleife**.

## 3.3 Schleifen: Aufgaben

```
for (int i = 1; i <= 50; i++)  
{  
    System.out.println(i);  
    i = i+7;  
}
```

### Aufgabe

1. Wie oft wird diese Schleife durchlaufen?
2. Welchen Wert hat i nach Abbruch der Schleife?

## 3.3 Schleifen: Aufgaben

```
for (int i = 1; i <= 50; i++)  
{  
    System.out.println(i);  
    i = i+7;  
}
```

```
1. i = 1  
2. i = 8  
3. i = 15  
4. i = 22  
5. i = 29  
6. i = 36  
7. i = 43  
8. i = 50
```

## 3.3 Schleifen: Aufgaben

```
for (int i = 1; i <= 10; i++)  
{  
    System.out.println(i);  
    i = i-2;  
}
```

### Aufgabe

Was läuft bei dieser Schleife falsch?



## 3.3 Schleifen: Aufgaben

```
for (int i = 1; i <= 10; i++)  
{  
    System.out.println(i);  
    i = i-2;  
}
```

### Aufgabe

Was läuft bei dieser Schleife falsch?

```
i = 1  
i = -1  
i = 0  
i = -2  
i = -1  
i = -3  
i = -2  
i = -4  
i = -3  
...
```

## 3.3 Schleifen: Aufgaben

```
for (int i = 1; i <= 100; i++)  
{  
    System.out.println(i);  
    i = i+2;  
}  
System.out.println(i);
```

### Aufgabe

Was läuft bei dieser Schleife falsch?

## 3.3 Schleifen: Aufgaben

```
for (int i = 1; i <= 100; i++)  
{  
    System.out.println(i);  
    i = i+2;  
}  
System.out.println(i);
```

### Aufgabe

Was läuft bei dieser Schleife falsch?

**Erstens:** Die Laufvariable wird im Schleifenrumpf verändert - kein guter Programmierstil. → kann Fehler verursachen.

**Zweitens:** Die Laufvariable ist außerhalb der Schleife nicht definiert → Fehler!

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #6

Schreiben Sie eine Methode

```
public getHaeufigkeitBuchstabe(String s, char c)
```

Diese Methode soll zählen, wie oft der angegebene Buchstabe c in dem String s vorkommt.

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #6

Schreibe eine Methode, die die Häufigkeit eines Buchstaben in einem String zählt.

```
public int getHaeufigkeitBuchstabe(String s, char c)
{
    int h = 0;

    for (int i=0; i<s.length(); i++)
        if (s.charAt(i) == c) h++;

    return h;
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #6

```
Schreibe eine Methode, die die Häufigkeit eines Buchstaben in einem String zählt.  
public int getHaeufigkeitBuchstabe(String s, char c)  
{  
    int h = 0;  
  
    for (int i=0; i<s.length(); i++)  
        if (s.charAt(i) == c) h++;  
  
    return h;  
}
```

Zwei wichtige Befehle der Klasse **String**:

- `length()` liefert die Anzahl der Zeichen.
- `charAt(p)` liefert das Zeichen an der Position `p`.

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #7

Berechnen Sie die Summe der ersten  $n$  Quadratzahlen, die durch drei teilbar sind und der ersten  $n$  Kubikzahlen, die durch fünf teilbar sind.

#### Beispiel:

Mit  $n = 10$  würden wir erhalten

$$3*3 + 5*5*5 + 6*6 + 9*9 + 10*10*10 = 1251$$

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #7

Berechnen Sie die Summe der ersten  $n$  Quadratzahlen, die durch drei teilbar sind und der ersten  $n$  Kubikzahlen.

#### Beispiel:

Mit  $n = 10$  würden wir erhalten

$$3*3 + 5*5*5 + 6*6 + 9*9 + 10*10*10 =$$

```
public int getSumme(int n)
{
    int sum = 0;

    for (int i=1; i<=n; i++)
        if (i % 3 == 0)
            sum += i*i;
        else if (i % 5 == 0)
            sum += i*i*i;

    return sum;
}
```



## 3.3 Schleifen: Aufgaben

### Übung 3.3 #8

Was macht folgende Schleife:

```
for (int a = 1; a <= 10; a++)  
{  
    for (int b = 1; b <= 10; b++)  
        System.out.printf("%6d", a*b);  
    System.out.println();  
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #8

Was macht folgende Schleife:

Ausgabe der Produkte  
von 1 x 1 bis 10 x 10

```
for (int a = 1; a <= 10; a++)  
{  
    for (int b = 1; b <= 10; b++)  
        System.out.printf("%6d", a*b);  
    System.out.println();  
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #9

Eine Methode erstellen, die einen Tannenbaum zeichnet.

```
      *  
    * * *  
  * * * * *  
* * * * * * *  
  * * * * * * * *  
* * * * * * * * *  
* * * * * * * * * *  
      *  
      *  
      *  
      *  
      *
```

## 3.3 Schleifen: Aufgaben

```
public void zeichne()
{
    for (int i=1; i<=7; i++)
    {
        for (int j=7-i; j>=1; j--)
            System.out.print(" ");

        for (int k=1; k<=i*2-1; k++)
            System.out.print("*");

        System.out.println();
    }

    for (int i=1; i<= 5; i++)
        System.out.println("
        *");
}
```

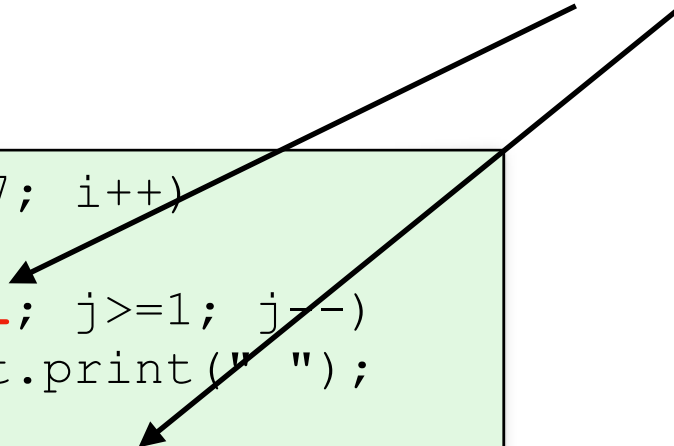
## 3.3 Schleifen: Aufgaben

Eine **innere** for-Schleife kann auf die Laufvariable der **äußeren** for-Schleife lesend zugreifen!

```
for (int i=1; i<=7; i++)
{
    for (int j=7-i; j>=1; j--)
        System.out.print(" ");

    for (int k=1; k<=i*2-1; k++)
        System.out.print("*");

    System.out.println();
}
```



## 3.3 Schleifen: Aufgaben

### Übung 3.3 #10

Eine Methode erstellen, die ein Schachbrettmuster erzeugt

```
# . # . # . # . # . # . #  
. # . # . # . # . # . # .  
# . # . # . # . # . # . #  
. # . # . # . # . # . # .  
# . # . # . # . # . # . #  
. # . # . # . # . # . # .  
# . # . # . # . # . # . #  
. # . # . # . # . # . # .
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #10

Eine M  
Schach

#.#.#.#.#.#.#.#

```
public void zeichneSchachbrett(int n)
{
    for (int zeile = 1; zeile <= n; zeile++)
    {
        for (int spalte = 1; spalte <= n; spalte++)
        {
            if ((zeile+spalte) % 2 == 0)
                System.out.print(". ");
            else
                System.out.print("# ");
        }
        System.out.println( );
    }
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #10

Eine Methode erstellen, die ein  
Schachbrettmuster erzeugt.

2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14
8	9	10	11	12	13	14	15
9	10	11	12	13	14	15	16

```
#.#.#.#.#.#.#.#  
.#.#.#.#.#.#.#.  
#.#.#.#.#.#.#.#  
.#.#.#.#.#.#.#.
```

```
#####  
#.  
#  
#.  
#
```

Die Werte aus (zeile + spalte).



## 3.3 Schleifen: Aufgaben

### Übung 3.3 #11

Erstellen Sie eine Klasse **Uhr** mit den Instanzvariablen **stunde** und **minute** (beide vom Typ `int`). Die Klasse soll folgende Methoden besitzen:

- `public void stellen(int h, int m)` - stellt die Uhrzeit auf die Stunde und Minute neu ein
- `public String getZeit()` - liefert einen String mit der Uhrzeit zurück, nach dem Muster `hh:mm`
- `public void anzeigen()` - zeigt den Uhrzeit-String in der Konsole an
- `public void tick()` - erhöht die Uhrzeit um genau eine Minute
- `public void run()` - lässt die Uhr in einer `for`- oder `while`-Schleife über 1500 Minuten laufen. Alle 10 Minuten soll die Uhrzeit in der Konsole ausgegeben werden.

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #11

Erstellen Sie eine Klasse **Uhr** mit den Instanzvariablen **stunde** und **minute** (beide vom Typ `int`). Die Klasse soll folgende Methoden besitzen:

- `public void stellen(int h, int m)` - stellt die Uhrzeit auf die Stunde und Minute neu ein
- `public String getZeit()` - liefert einen String mit der Uhrzeit in der Form hh:mm
- `public void anzeigen()` - zeigt den Uhrzeit-String in der Konsole an
- `public void tick()` - erhöht die Uhrzeit um genau eine Minute
- `public void run()` - lässt die Uhr in einer for- oder while-Schleife über 1500 Minuten laufen. Alle 10 Minuten soll die Uhrzeit in der Konsole ausgegeben werden.

Lösung siehe laufendes Programm!

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #12

Schreiben Sie eine Methode `public void iteriere()`

Zunächst soll die Methode mit `double zahl = Math.random() * 10` eine Zufallszahl zwischen 0 und 10 (exklusiv) erzeugen.

Dann soll folgender Algorithmus in einer Schleife wiederholt werden:

- Falls  $zahl < 1$ , dann multipliziere Zahl mit 2
- Falls  $zahl \geq 1$ , dann dekrementiere Zahl um 1.

Die Schleife soll beendet werden, sobald  $Zahl \leq 0$  ist oder mehr als 100 Durchläufe erfolgt sind.

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #12

```
public void iteriere()  
{  
    double zahl = Math.random() * 10;  
    int i = 1;  
    System.out.printf("Gewählte Zahl: %6.3f %n%n", zahl);  
  
    while (zahl > 0 && i <= 100)  
    {  
        if (zahl < 1)  
            zahl *= 2;  
        else if (zahl >= 1)  
            zahl--;  
        System.out.printf("i = %4d, zahl = %6.3f %n", i, zahl);  
        i++;  
    }  
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #13

Schreiben Sie eine Methode ähnlich wie in 3.3 #12, die die COLLATZ-Folge ausgibt:

- Falls zahl gerade,  $\text{zahl} \rightarrow \text{zahl} / 2$
- Falls zahl ungerade:  $\text{zahl} \rightarrow 3 * \text{zahl} + 1$ .

Beispiel für  $\text{zahl} = 7$ :

7 - 22 - 11 - 34 - 17 - 52 - 26 - 13 - 40 - 20 - 10 - 5 - 16 - 8 - 4 - 2 - 1

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #13

Schreiben Sie eine Methode ähnlich wie in 3.3 #12, die die COLLATZ-Folge ausgibt:

- Falls zahl
- Falls zahl

Beispiel für

7 - 22 - 11

```
public void iteriereW(int zahl)
{
    System.out.printf("Startwert = " + zahl);
    int i = 1;

    while (i <= 100)
    {
        if (zahl % 2 == 0) zahl /= 2;
        else zahl = 3*zahl + 1;
        System.out.printf("i = %4d, zahl = %4d %n", i, zahl);
        if (zahl == 1) break;
        i++;
    }
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #13

Schreiben Sie eine Methode ähnlich wie in 3.3 #12, die die COLLATZ-Folge ausgibt:

- Falls zahl
- Falls zahl

Beispiel für

7 - 22 - 11

```
public void iteriereF(int zahl)
{
    System.out.printf("Startwert = " + zahl);

    for (int i=1; i <= 100; i++)
    {
        if (zahl % 2 == 0) zahl /= 2;
        else zahl = 3*zahl + 1;
        System.out.printf("i = %4d, zahl = %4d %n", i, zahl);
        if (zahl == 1) break;
    }
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #14

Schreiben Sie eine Methode, die überprüft, ob die übergebene Zahl eine Fakultät ist.

```
public void testeAufFakultaet(int zahl)
```

#### Beispiele:

zahl = 362880, Ausgabe der Methode: 362880 ist 9!

zahl = 1000, Ausgabe der Methode: 1000 ist keine Fakultät.



## 3.3 Schleifen: Aufgaben

### Übung 3.3 #14

Schreiben Sie eine Methode, die überprüft, ob die übergebene Zahl eine

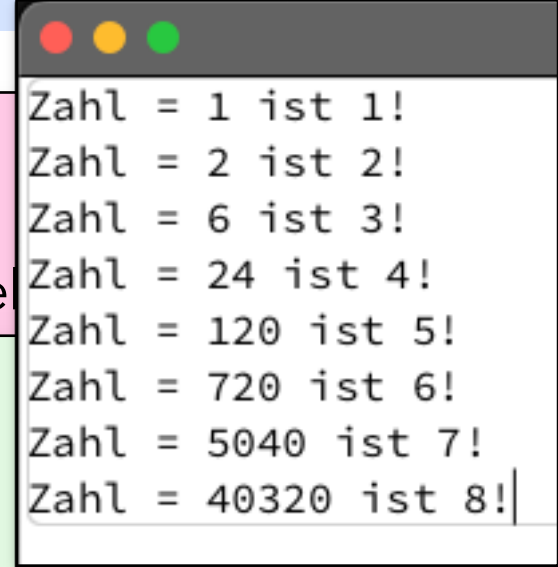
```
public void testeAufFakultaet(int zahl)
{
    for (int i=1; i<=14; i++)
    {
        if (fakultaet(i) == zahl)
        {
            System.out.println("Zahl = "+zahl+" ist " + i+"!");
            break;
        }
    }
    System.out.println("Zahl = "+zahl+" ist keine Fakultät");
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #14

Schreiben Sie eine Methode, die überprüft, ob die übergebene Zahl eine Fakultät ist.

```
public void testeAufFakultaet(int zahl)
{
    for (int i=1; i<=14; i++)
    {
        if (fakultaet(i) == zahl)
        {
            System.out.println("Zahl = "+zahl+" ist " + i+"!");
            break;
        }
    }
    //System.out.println("Zahl = "+zahl+" ist keine Fakultät");
}
```



```
Zahl = 1 ist 1!
Zahl = 2 ist 2!
Zahl = 6 ist 3!
Zahl = 24 ist 4!
Zahl = 120 ist 5!
Zahl = 720 ist 6!
Zahl = 5040 ist 7!
Zahl = 40320 ist 8!
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #15, Teil 1

Schreiben Sie dann eine Methode

```
public String getDual(int dezimal)
```

Diese Methode nimmt eine int-Zahl entgegen und liefert einen String aus Nullen und Einsen zurück, beispielsweise "100101001", der die Dualdarstellung der Dezimalzahl repräsentiert.

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #15, Teil 1

```
Sch public String getDual(int dezimal)
{
    if (dezimal == 0) return "0";

    String dual = ""; // klein fängt man an...

    while (dezimal > 0)
    {
        int rest = dezimal % 2; // Rest bei Division durch 2
        dual      = rest + dual; // vorn (!) anfügen
        dezimal   = dezimal / 2; // Zahl halbieren
    }

    return dual;
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #15, Teil 2

Schreiben Sie dann eine Methode

```
public int getDezimal(String dual)
```

Diese Methode wandelt den String dual wieder in eine Dezimalzahl um.

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #15, Teil 2

Schreiben Sie dann ein

```
public int getDe
```

Diese Methode wand

```
public int getDezimal(String dual)
{
    int dezimal = 0;

    for (int i = 0; i < dual.length(); i++)
    {
        char c = dual.charAt(i);

        if (c == '1')
            dezimal = dezimal * 2 + 1;
        else if (c == '0')
            dezimal = dezimal * 2;
        // andere Zeichen werden ignoriert
    }
    return dezimal;
}
```

## 3.3 Schleifen: Aufgaben

### Übung 3.3 #15, Teil 2

Schreiben Sie dann ein

pub

Dies

```
public boolean testOK(int zahl)
{
    String dual = getDual(zahl);
    System.out.println(dual);

    int dezimal = getDezimal(dual);
    System.out.println(dezimal);

    return dezimal == zahl;
}
```

```
public int getDezimal(String dual)
{
    for (int i = 0; i < dual.length(); i++)
    {
        char c = dual.charAt(i);
        if (c != '0')
        {
            dezimal * 2 + 1;
        }
        else
        {
            dezimal * 2;
        }
    }
    return dezimal;
}
```