

# Programmiervorkurs

## Wintersemester 20/21

Tag 4



# Agenda

- › Aufgaben zur Wiederholung
- › Tutorium
  - do-while-Schleife
  - while-Schleife
  - for-Schleife
- › Mäxchen 4.0 & Aufgabe



# Problemstellung

› Bereits bekannt von Tag 2: Menü, das Benutzer verschiedene Optionen wählen lässt

› Menü wird nur einmal angezeigt und Benutzer kann nur eine Option wählen. Danach ist das Programm beendet.

1. Hallo ausgeben  
2. Zahl quadrieren  
3. Zahl größer 0 testen  
4. Programmende ausgeben

Programmfunktion? 3

Welche Zahl soll geprüft werden? 43

Zahl ist größer 0!

**Schleifen!**

→ gesucht: Möglichkeit, das Menü öfter auszugeben und das Programm erst dann zu beenden, wenn der Benutzer das möchte.

# do-while-Schleife

› fußgesteuerte Schleife

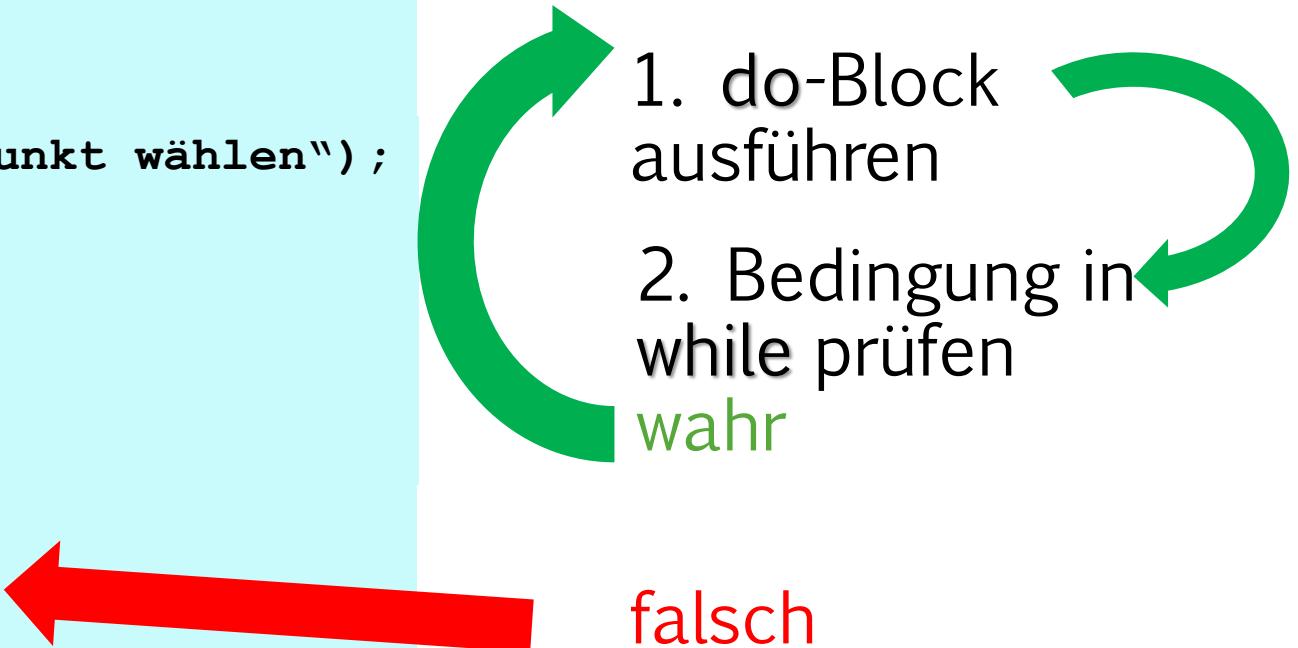
```
...
int choice = 0;
do{
    System.out.println("Menüpunkt wählen");
    ...
    choice = sc.nextInt();
    ...
    switch/case
    ...
} while( choice != 4 );
...
```

Ablauf:

1. do-Block  
ausführen

2. Bedingung in  
while prüfen  
wahr

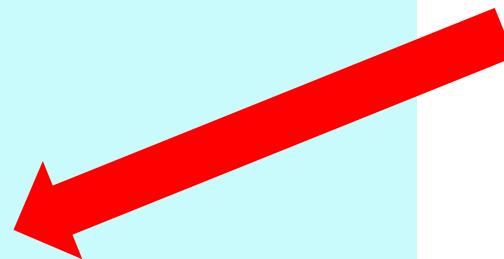
falsch  
→ es geht unter  
der  
while weiter



# while-Schleife

› kopfgesteuerte Schleife

```
...  
int i = 0;  
while( i < 100 ) {  
    System.out.println(i);  
    i++;  
}  
...
```



Ablauf:

1. Bedingung  $i < 100$  while prüfen
- 
- falsch → es geht unter der while weiter
- wahr Code Block ausführen

# Aufgabe: Schleifen



## GRUNDLAGE (PABS)

- › Schreibt eine Methode `zaehleBisZahlUndRufe(int zahl)`, der ihr einen int übergeben könnt und die euch alle Zahlen bis zu dieser Zahl und am Ende ein “Ich komme!” ausgibt.
- › Schreibt eine Methode `fakultaet`, der ihr einen int übergeben könnt und der Euch die Fakultät dieses Wertes berechnet

## ZUSATZ

- › Erstellt ein Programm, das die Kreiszahl  $\pi$  nach der Reihenentwicklung von Leibniz berechnet:
- ›  $\pi/4 = 1/1 - 1/3 + 1/5 - 1/7 + 1/9 \dots$
- › Die Schleife soll laufen, bis das Ergebnis 3,14159XXX beträgt.
- › Gib dann auf der Konsole aus, wie oft die Schleife gelaufen ist.

# for-Schleife

› Verkürzung der while-Schleife

```
...
int i=0;
while( i < 100 ){
    System.out.println(i);
    i++;
}
...
```

1. Initialisierung

2. Bedingung ja/nein

3. Code-Block

4. Inkrement

```
...
for( int i=0 ; i < 100 ; i++ ){
    System.out.println(i);
}
...
```

# Aufgabe: Mehr Schleifen



## GRUNDLAGE (PABS)

- › Schreibt eure Methode `zaehleBisZahlUndRufe` so um, dass sie das Gleiche tut, aber eine `for`-Schleife benutzt.
- › Schreibt eine Methode, die die Zahlen von 50 bis -100 ohne die 0 auf der Konsole ausgibt.

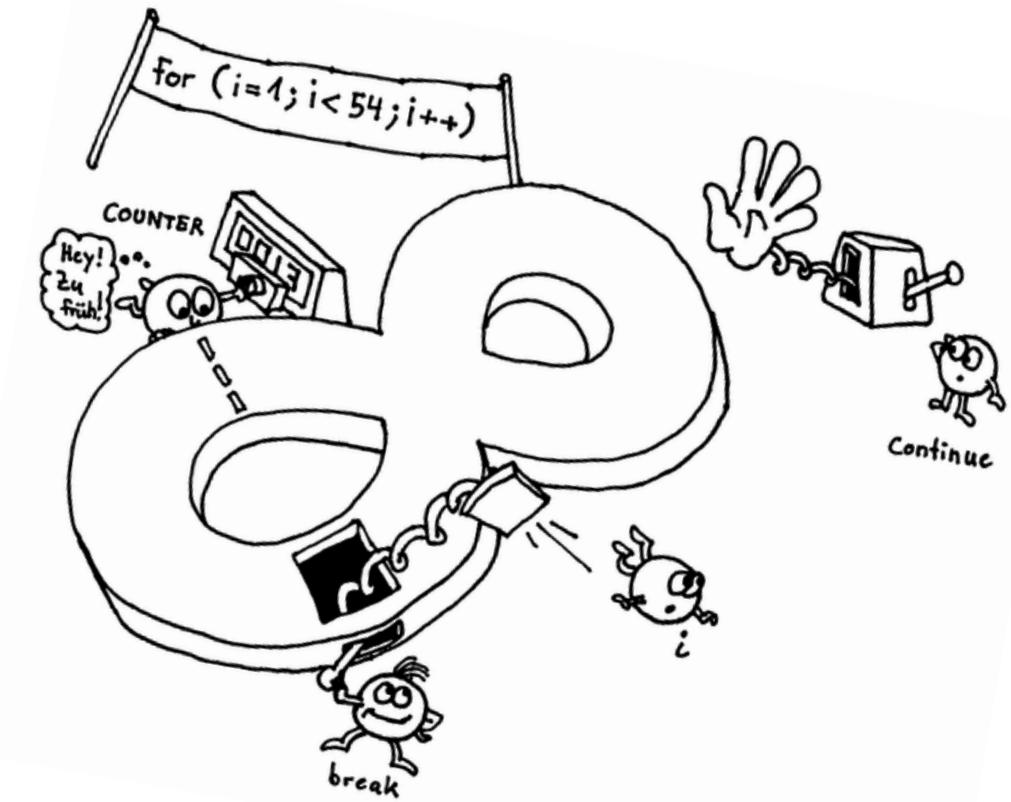
## ZUSATZ

- › Erstellt ein Programm, das die Kreiszahl  $\pi$  durch das zufällige Streuen von Punkten in ein Quadrat ermittelt.
- › Verwendet als Zufallsgenerator die Methode `Math.random()`.
- › Hinweise zu dieser Variante findet Ihr im Internet unter dem Schlagwort “Monte-Carlo-Methode”.

# Endlosschleife

## › Beispiel: while-Schleife

```
...  
  
int i=0  
  
while( true ) {  
  
    System.out.println(i);  
    i++;  
  
}  
  
...
```



Bedingung ist immer wahr!  
→ Schleife bricht nie ab

# Mehrfachschleifen

- › gesucht: Programm, das Folgendes ausgibt:

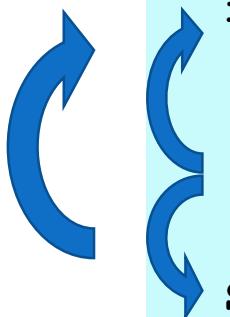
1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
...									
...									

```
for (int zeile = 1; zeile <= 10; zeile++) {  
  
    for(int spalte = 1; spalte <= 10; spalte++) {  
        int produkt = zeile * spalte;  
        System.out.print(produkt + "\t");  
    }  
  
    System.out.print("\n");  
}
```

# Mehrfachschleifen

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
...									
...									

```
for (int zeile = 1; zeile <= 10; zeile++) {  
  
    for(int spalte = 1; spalte <= 10; spalte++) {  
        int produkt = zeile * spalte;  
        System.out.print(produkt + "\t");  
    }  
    System.out.print("\n");  
}
```



# Aufgabe: Schon wieder Schleifen

# GRUNDLAGE

- › Erstellt ein Programm, dass vom Nutzer ein beliebiges Zeichen einliest sowie eine Ganzzahl  $n$ . Das Zeichen soll  $n$  Zeilen oft  $n$ -mal ausgegeben werden.

## ZUSATZ

- › Erstellt ein Programm, das (mit Hilfe von for-Schleifen) symmetrische Tannenbäume (variabler Höhe) generiert. Der Stamm hat immer eine Höhe von zwei Zeilen und die Krone mindestens drei.

### Höhe 3

## Höhe 5

# Übungen aus der Aufgabensammlung

Name	Seite	Schwierigkeit (1-10)	Freiheit (0-3)	Themen
Folgen Erkennen	14	3	1	Schleifen
Würfeln	15	4	1	Schleifen, [Objekte/Methoden] (optional)
Harmonische Reihe	16	4	1	Schleifen, Mathe
Menü 2	13	2	0	Schleifen, Bedingte Anweisungen
Bäume zeichnen	24	7	3	Schleifen, Logik
Monte-Carlo (Pi berechnen)	28	7	3	Logik, Mathe, Schleifen

# Abschlussaufgabe: Autohaus

- › Erstellt eine Klasse “Auto“, welche die folgenden Attribute hat:  
String modell, String farbe, int alter
- › Erstellt eine weitere Klasse “Autohaus“, die ein Objekt vom Typ “Auto“ anlegt, anschließend die Werte des Autos einliest und in dem Objekt speichert.
- › Der Benutzer wird nach jedem Auto gefragt, ob er noch ein weiteres Auto einlesen möchte. Falls “ja“, wird von vorne mit Einlesen begonnen
- › Bei der Eingabe des nächsten Autos wird das vorherige Auto durch Neuanlage des Objektes überschrieben (einen “Speicher“ für mehrere Autos implementieren wir später)
- › Zum Schluss soll das aktuelle Auto ausgegeben werden