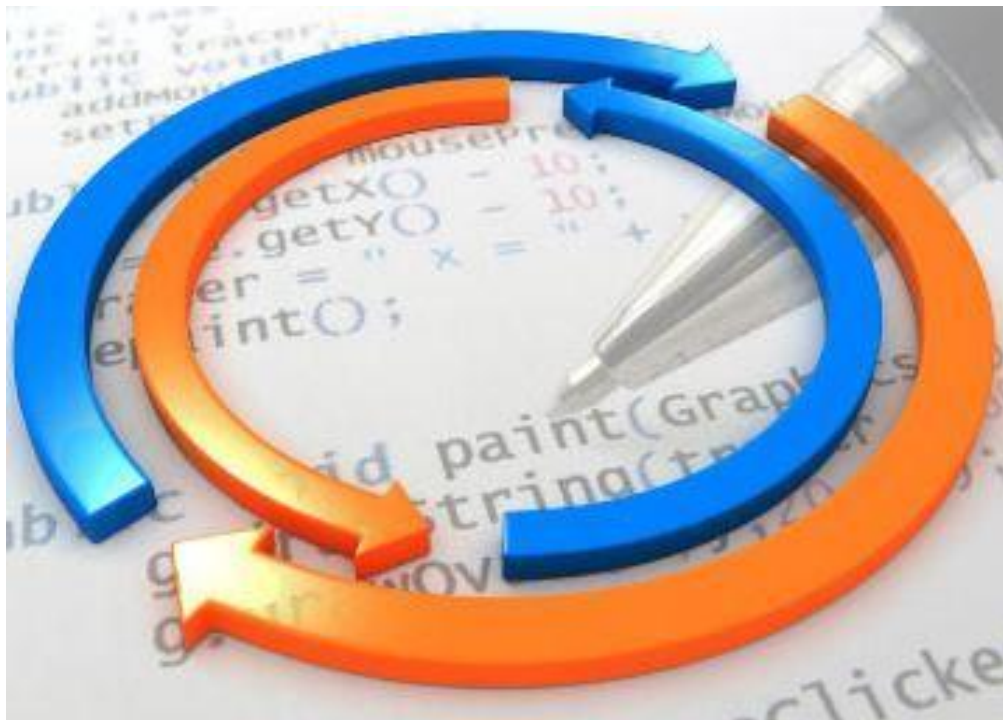


AP-10

Zufallszahlen, Schleifen und Arrays

Die Wiederholung von Anweisungen und
die Anordnung von Daten des gleichen Datentyps



Stand: **01.01.2020**
Autor: **M. Völkl**

1. Zufallszahlen



Definition

Als Zufallszahl wird das Ergebnis von speziellen Zufallsexperimenten bezeichnet. Zur Erzeugung von Zufallszahlen gibt es verschiedene Verfahren. Diese werden als **Zufallszahlengeneratoren** bezeichnet.

Beispiel

Durch `Math.random()` werden zufällige Kommazahlen vom Datentyp **double** erzeugt.

Die erzeugten Zufallszahlen besitzen **Werte zwischen 0 und minimal kleiner als 1**.

```
System.out.println(Math.random());  
System.out.println(Math.random());
```

Ausgabe:

```
0.8215487886657413  
0.3279285667938139
```

Anwendung

Oftmals werden keine Fließkommazahlen gesucht, sondern ganze Zahlen.

Eine ganzzahlige Zufallszahl zwischen **0** und **4** wird mit der unten dargestellten Programmzeile erzeugt.

Wir deklarieren eine **int-Variable**, da die Zufallszahl eine ganze Zahl sein soll.

Eine **Klasse**, die mit Java ausgeliefert wird.

```
int zufallsZahl = (int) (Math.random() * 5) ;
```

Das ist eine **Typumwandlung (cast)**, der die unmittelbar darauf folgende Variable in einen Integer Wert umwandelt. Der Bereich rechts vom Komma wird abgeschnitten. Bei der erzeugten Zahl 4.9999 bleibt nur die Zahl 4 zurück.

Eine **Methode** der Klasse `Math`.

Die Methode `Math.random` liefert eine Zahl zwischen 0 und irgendetwas minimal kleiner als 1 zurück.

Durch die Multiplikation mit 5 ist die höchste erzeugte Zufallszahl 4.99999...

Aufgaben

Bestimmen Sie den möglichen Ausgabenbereich für die Variable **zufallsZahl**.

`int zufallsZahl = (int) (Math.random()*3);`

`int zufallsZahl = (int) (Math.random()*1) ;`

`int zufallsZahl = (int) (Math.random()*-2) ;`

`int zufallsZahl = (int) (Math.random()*3 + 1) ;`

`int zufallsZahl = (int) (Math.random()*3 + 2) ;`

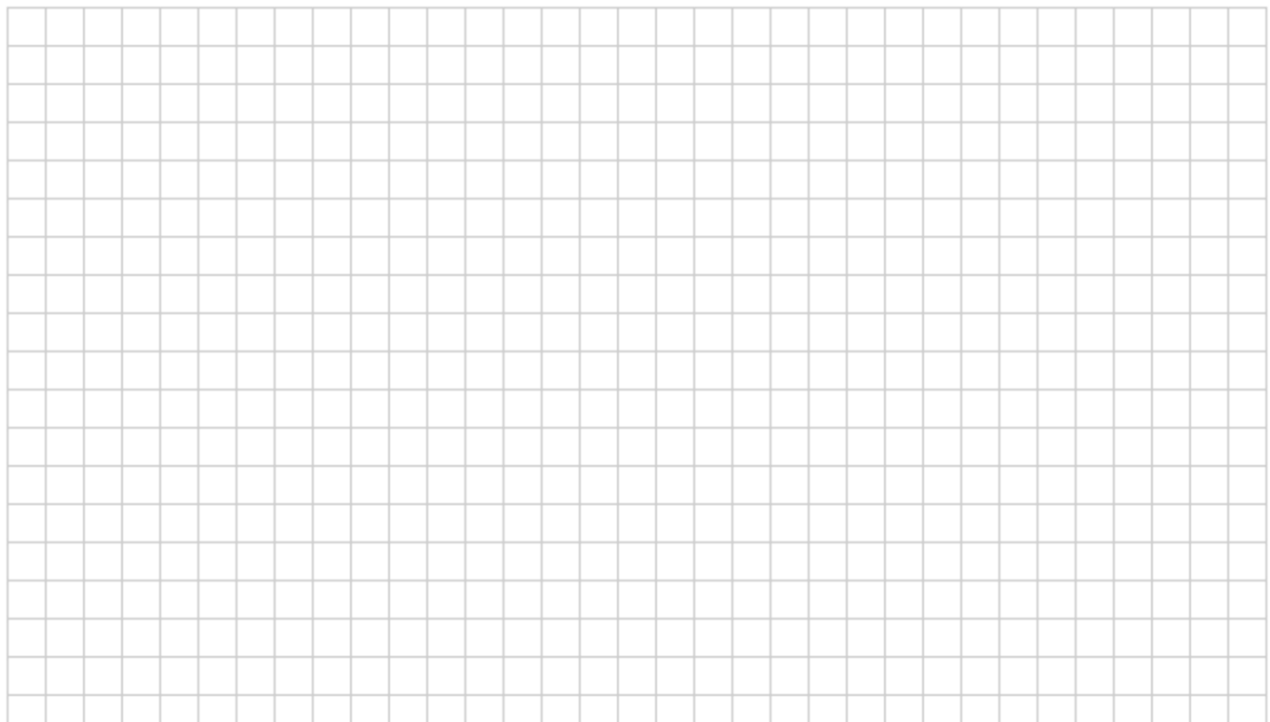
`int zufallsZahl = (int) (Math.random()*(-2) - 2) ;`



Programmierauftrag

Schreiben Sie ein Programm das den Würfelwurf simuliert. **Falls** eine Sechs gewürfelt wird erscheint die Meldung Gewinn, **sonst** die Meldung Niete. Erstellen Sie dazu zuerst das **Struktogramm**!

Erweitern Sie anschließend Ihr Programm. Es werden zwei gleichzeitige Würfe eines Würfels simuliert. Falls es sich um einen Pasch handelt (die Würfel zeigen die gleiche Zahl) erscheint die Meldung Gewinn, sonst die Meldung Niete.



2. Schleifen

Eine **Schleife** ist eine Kontrollstruktur in Programmiersprachen. Sie wiederholt einen Anweisungs-Block – den sogenannten *Schleifenkörper* – solange die **Schleifenbedingung** gültig bleibt bzw. die Abbruchbedingung nicht eintritt. Schleifen, deren Schleifenbedingung immer gültig bleibt, sind *Endlosschleifen*.

Grundsätzlich lassen sich *drei Schleifenarten* unterscheiden.

- **zählergesteuerte Schleife (For Schleife)**

Bei dieser Schleife steht die Anzahl der Schleifenläufe bereits im Vorfeld fest. Eine Zählvariable wird hier eine bestimmte Anzahl oft durchlaufen.

- **kopfgesteuerte Schleife (While Schleife)**

Diese Schleife wird solange durchlaufen, solange die Bedingung erfüllt ist.

- **fußgesteuerte Schleife (Do-While Schleife)**

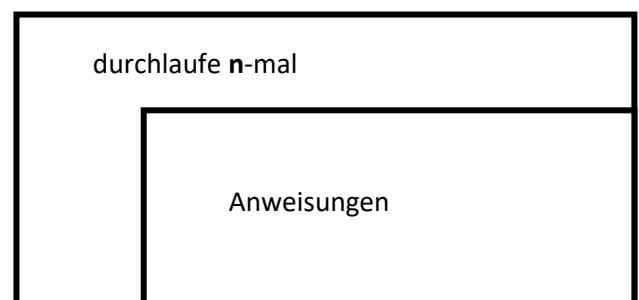
Diese Schleife wird nur durchlaufen, solange die Bedingung erfüllt ist, mindestens jedoch einmal.

2.1 Die *For* Schleife – Eine zählergesteuerte Schleife

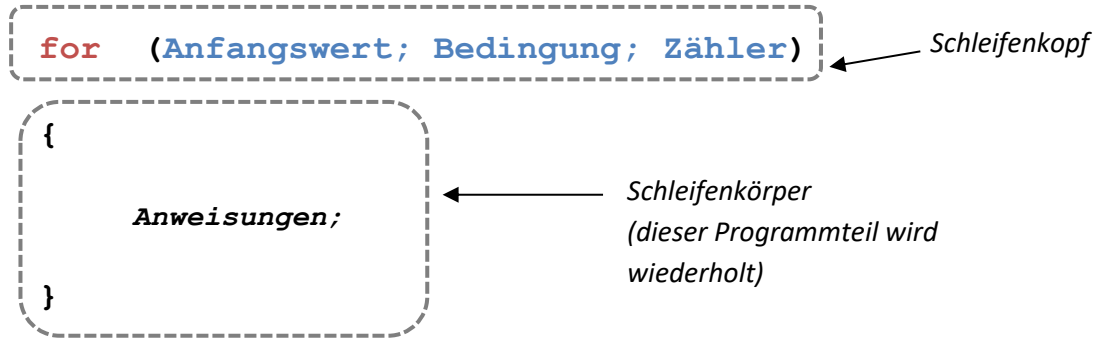
Definition

Die For–Schleife ist eine **Struktur zur Wiederholung von Anweisungen**. Die Anzahl der Schleifendurchläufe steht schon beim Eintritt in die Schleife fest. Sie wird sooft durchgelaufen, bis ein **Zähler (Zählvariable)** ausgehend von einem **Startwert** ihren **Endwert** erreicht hat. Dabei verändert sich die Zählvariable bei jedem Durchgang immer um den gleichen, festgelegten Wert. Da der Durchlauf durch eine Zählvariable gesteuert wird, bezeichnet man diese Schleife auch als **zählergesteuerte Schleife**.

Das **Struktogramm** zeigt das Ablaufdiagramm für das Durchlaufen einer Schleife. Es muss erkenntlich gemacht werden, wie oft die Anweisungen ausgeführt werden (hier **n-mal**).



Aufbau



- Der **Anfangswert** legt den Startwert fest, ab dem gezählt wird. Die Zählvariable (meist *i*) wird hier *deklariert* und *initialisiert*.
- Die **Bedingung** legt den Endwert fest, bis zu dem gezählt werden soll.
- Der **Zähler** gibt an, wie sich die Zählvariable bei jedem Durchgang ändert.

Beispielcode

```

for (int i = 0; i<=4; i++) // i++ bedeutet i = i + 1;
{
    System.out.println(i);
}

```

Ausgabe:

```

0
1
2
3
4

```

Im obigen Beispiel wird die Schleife insgesamt **5-mal** durchlaufen. Der erste Wert für die Zählvariable liegt bei **0**, der letzte Wert bei **4**, da durch das **<=** Zeichen die 4 mit eingeschlossen ist.

Abbruchkriterium

Die **Bedingung** im Schleifenkopf ist entscheidend für den Durchlauf und Abbruch der Schleife. Bei einem **kleiner** Zeichen **<** ist die letzte Zahl **nicht** mit eingeschlossen, während bei einem **kleiner gleich** Zeichen **<=** die letzte Zahl **noch** mit dabei ist.



Welche Ausgabe liefern die folgenden Programmcode-Zeilen?

a)

```

for (int i=0; i<2; i++){
    System.out.println(i);
}

```

b)

```

for (int i=0; i<3; i+=2){
    System.out.println(i);
}

```

Ausgabe:

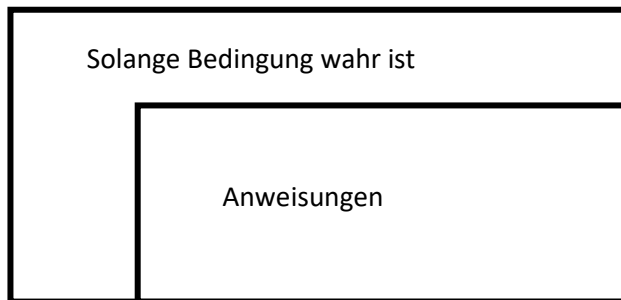
Ausgabe:

2.2 Die *While* Schleife – Eine Kopfgesteuerte Schleife

Definition

Bei einer kopfgesteuerten Schleife erfolgt die Abfrage der Bedingung **bevor** der Schleifenrumpf ausgeführt wird, also am *Kopf* des Konstrukts. Solange diese Bedingung wahr ist, werden die Anweisungen innerhalb des Schleifenkörpers ausgeführt.

Struktogramm



Code

```

while (Bedingung)
{
    Anweisung;
}
  
```

Beispielcode

a)

```

int i = 0;
while (i<=3) {
    System.out.println(i);
    i++;
}
  
```

Ausgabe:

0
1
2
3

b)

```

int i = 0;
while (i<3) {
    i++;
    System.out.println(i);
}
  
```

Ausgabe:

1
2
3



Welche Ausgabe liefern die folgenden Programmcode-Zeilen?

a)

```

int i = 1;
while (i<4) {
    i++;
    System.out.println(i);
}
  
```

Ausgabe:

b)

```

int i = 0;
while (i<4) {
    i+=2;
    System.out.println(i);
}
  
```

Ausgabe:

2.3 Die *do-while* Schleife – Eine fußgesteuerte Schleife

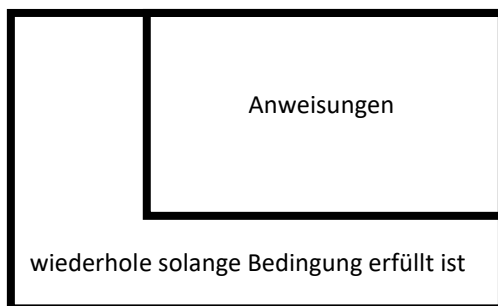
Definition

Die *do-while* Schleife wird solange durchlaufen, wie die Bedingung erfüllt ist. **Auf jeden Fall wird die Schleife einmal durchlaufen** (auch wenn die Bedingung nicht erfüllt ist), da die Abbruchbedingung **erst nach den Anweisungen geprüft** wird.



Die *do-while*-Schleife wird mit einem Strichpunkt (;) abgeschlossen! Die Variable(n) in der Bedingung muss (müssen) zuvor deklariert und initialisiert werden!

Struktogramm und Pseudocode



```
do
{
    Anweisungen;
}
while (Bedingung);
```

Beispielcode

```
int i = 0;
do {
    System.out.println(i);
    i++;
} while (i < 3);
```

Ausgabe:
0
1
2

```
int i = 0;
do {
    i++;
    System.out.println(i);
} while (i < 3);
```

Ausgabe:
1
2
3



Welche Ausgabe liefern die folgenden Programmcode-Zeilen?

a)

```
int i = 0;
do {
    System.out.println(i);
    i++;
} while (i <= 3);
```

Ausgabe:

b)

```
int i = 1;
do {
    System.out.println(i);
    i+=2;
} while (i <= 5);
```

Ausgabe:

Auftragsstellung

Die Firma *Javagames – Coffee and Games* möchte ein paar Konsolenprogramme entwickeln, um Admins bei längeren Installationsaufgaben und Wartezeiten etwas Entspannung zu gönnen.



Sie sollen die **Programme** federführend in Ihrer Abteilung **entwickeln** und Ihren Kolleginnen und Kollegen bei den wichtigsten **Fragestellungen** helfen.

Anschließend sollen Sie die entwickelten Programme vorstellen!

Informieren Sie sich mithilfe des **Schleifen Kompendiums** und weiteren Online-Quellen (Tutorials, Howto's).



Fragestellungen

1. Es gibt drei verschiedene Arten von Schleifen. Nennen Sie diese!

2. Wann findet die Prüfung der Bedingungen bei der kopf- und der fußgesteuerten Schleife statt?

3. Beschreiben Sie in Stichpunkten, was man unter dem Schleifenkörper versteht!

4. Wie lange ist die Zählvariable (z.B. *i*) in der for-Schleife gültig bzw. sichtbar?

5. Wie oft muss eine kopf- /fußgesteuerte Schleife mindestens durchlaufen werden?

Theorie-Aufgaben



Ihre Kolleginnen und Kollegen haben bereits zwei Programme entwickelt.

Analysieren Sie die Programme und bestimmen Sie die benötigten Werte von i sowie die Schleifendurchläufe!

Wie oft werden die einzelnen Schleifen durchlaufen?

```
for (int i = 0; i<=3; i++)
{
    System.out.println(i) ;
}
```

Erster Wert für i : _____

Letzter Wert für i in der Schleife: _____

Anzahl der Schleifendurchläufe: _____

```
for (int i = 1; i<5; i+=2)
{
    System.out.println(i) ;
}
```

Erster Wert für i : _____

Letzter Wert für i in der Schleife: _____

Anzahl der Schleifendurchläufe: _____

Gesuchter
Wert

Lösungs-
Buchstabe

0	Z
1	S
3	Ä
4	C
3	T
4	H
0	T
1	L
2	A
3	E
2	R
3	H

Lösungswort: _____



Programmierauftrag 1 – Gewinn oder Niete

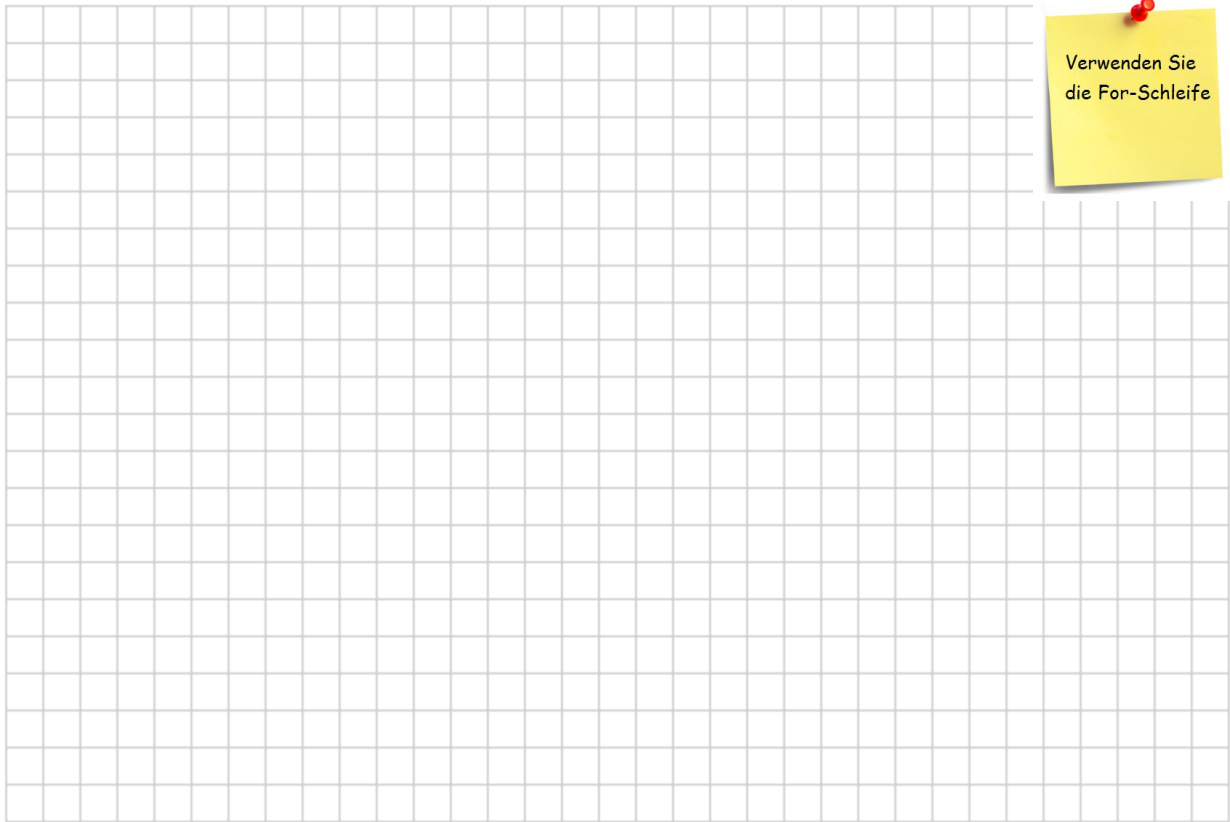


Entwickeln Sie ein Programm, das den Wurf eines Würfels simuliert und die geworfene Zahl ausgibt. **Falls** eine 6 gewürfelt wird erscheint die Meldung *Gewinn*, **ansonsten** die Meldung *Niete*. **Der Spieler würfelt in jedem Fall 10-mal.**

Für Experten: Die Anzahl der Gewinne und die Anzahl der Nieten soll ausgegeben werden.



Erstellen Sie das Struktogramm zur Lösung des dargestellten Problems.



Verwenden Sie
die For-Schleife



Erstellen Sie ein Java-Programm zur Lösung der dargestellten Problemstellung!

Achten Sie auf eine strukturierte Darstellung des Programmes und dokumentieren Sie wichtige Inhalte mithilfe der Kommentarfunktion. Speichern und/oder drucken Sie Ihr Programm aus.

Programmierauftrag 2 – Zahlen raten



Es soll eine Zufallszahl zwischen **1** und **10** generiert werden.

Anschließend wird der Benutzer aufgefordert eine Zahl zwischen 1 und 10 einzugeben, um die Zufallszahl **zu erraten**.

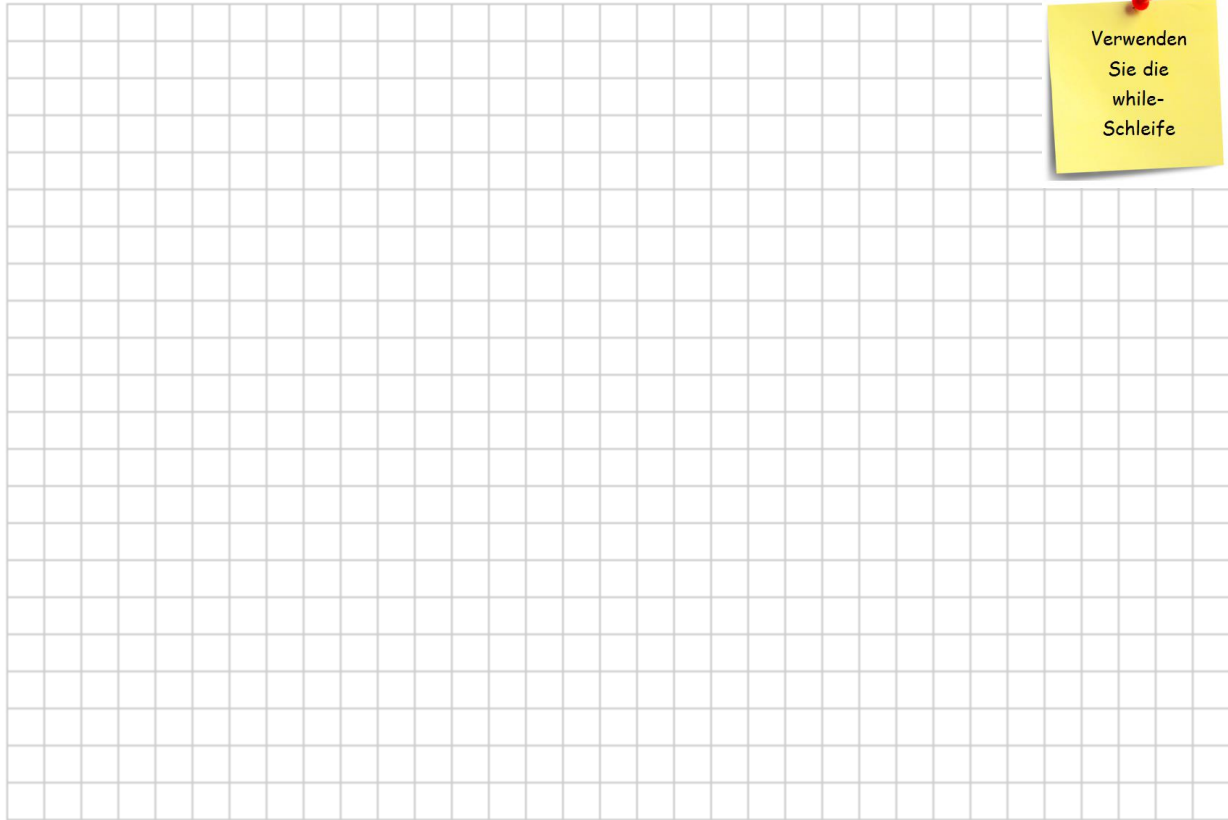
Solange eine falsche Zahl eingegeben wird, erscheint die Meldung *Falsch* und der Benutzer soll einen neuen Versuch erhalten.

Nach dem Erraten der Zahl erscheint die Meldung *Richtig* und das Programm wird beendet.

Für Experten: Die Anzahl der Versuche soll ausgegeben werden.



Erstellen Sie das Struktogramm zur Lösung des dargestellten Problems.



Verwenden
Sie die
while-
Schleife

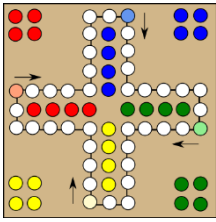
fsi_Schleifen_Vk_20200104.docx



Erstellen Sie ein Java-Programm zur Lösung der dargestellten Problemstellung!

Achten Sie auf eine strukturierte Darstellung des Programmes und dokumentieren Sie wichtige Inhalte mithilfe der Kommentarfunktion. Speichern und/oder drucken Sie Ihr Programm aus.

Programmierauftrag 3 – *Mensch ärgere dich nicht!*



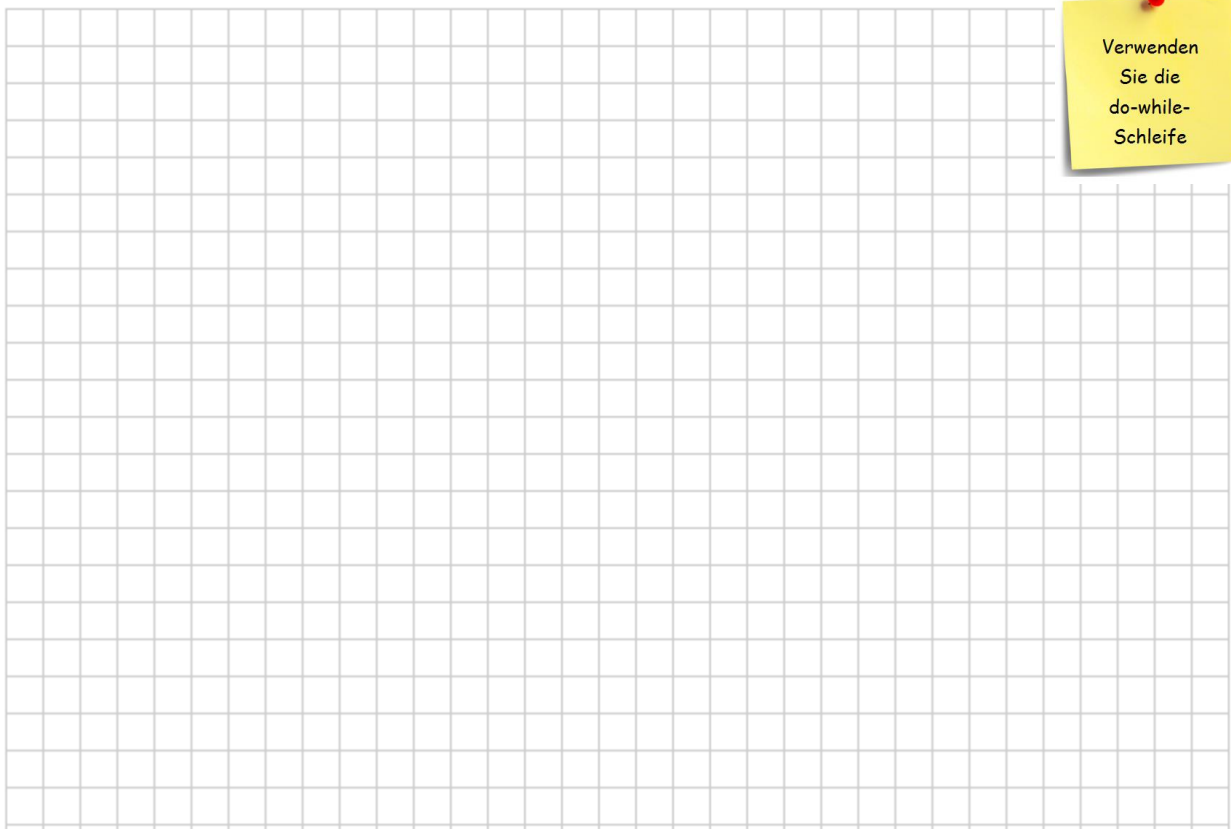
Erstellen Sie ein Programm, das den Wurf eines Würfels simuliert und die geworfene Zahl ausgibt.

Der Spieler darf nochmal würfeln, falls er eine 6 hat.

Für Experten: Geben Sie am Ende aus, wie oft der Spieler gewürfelt hat.



Erstellen Sie das Struktogramm zur Lösung der dargestellten Problematik!



Verwenden
Sie die
do-while-
Schleife

fisi_Schleifen_Vk_20200104.docx



Erstellen Sie ein Java-Programm zur Lösung der dargestellten Problemstellung!

Achten Sie auf eine strukturierte Darstellung des Programmes und dokumentieren Sie wichtige Inhalte mithilfe der Kommentarfunktion. Speichern und/oder drucken Sie Ihr Programm aus.

Kontrollfragen

Aufgabe 1

Welchen Wert besitzt i nach dem Verlassen der Schleife?

a)

```
int i = 1;
do {
    System.out.println(i);
    i++;
} while(i<=3);
```

Wert: _____

b)

```
int i = 1;
while (i<=5) {
    System.out.println(i);
    i+=2;
}
```

Wert: _____

Aufgabe 2

Wie oft werden die einzelnen Schleifen durchlaufen?

```
for (int i = 1; i>0; i++)
{
    System.out.println(i);
}
```

Anzahl der Schleifendurchläufe: _____

```
for (int i = 1; i<4; i=i+4)
{
    System.out.println(i);
}
```

Anzahl der Schleifendurchläufe: _____

```
int i = 0;
while (i<=5) {
    i=i+2;
    System.out.println(i);
}
```

Anzahl der Schleifendurchläufe: _____

```
int i = 3;
do {
    System.out.println(i);
    i++;
} while(i<3);
```

Anzahl der Schleifendurchläufe: _____

Lösungs-
Buchstabe

Gesuchter
Wert

0 H
∞ p

0 u
1 i

2 b
3 n

0 d
1 g

Lösungswort: _____

Aufgaben für Experten (optional)

Experten-Aufgabe 1: Zahlenraten

Schreibe ein einfaches Ratespiel. Der Spieler soll eine Zufallszahl erraten die zwischen 0 und einer vom Spieler gewählten Obergrenze liegt. Der Spieler darf einen Tipp abgeben. Das Programm antwortet jeweils ob die getippte Zahl kleiner oder größer als die Zufallszahl ist. Das Spiel läuft solange bis der Spieler die Zahl erraten hat. Zusätzlich soll die Anzahl der Versuche gezählt werden.

Der Spielablauf soll also in etwa wie folgt aussehen:

```
Bitte gebe eine Obergrenze an: 50
Ok, los geht's!
Versuch 1 - Dein Tipp? 25
Meine Zahl ist groesser!
Versuch 2 - Dein Tipp? 40
Meine Zahl ist kleiner!
Versuch 3 - Dein Tipp? 30
Meine Zahl ist groesser!
Versuch 4 - Dein Tipp? 35
Meine Zahl ist kleiner!
Versuch 5 - Dein Tipp? 32
Meine Zahl ist kleiner!
Versuch 6 - Dein Tipp? 31
Du hast die Zahl mit 6 Versuchen erraten!
```

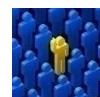
Experten-Aufgabe 2: Würfelspiel erweitern



Erweitern Sie Ihren Programmierauftrag 1: *Gewinn oder Niete*

Der Benutzer darf die *maximale Anzahl an Würfeln bestimmen*. Falls eine Sechs gewürfelt wird, wird *Gewinn* angezeigt und *die Schleife vorzeitig verlassen*. Abbrüche der Schleife können mit dem Befehl **break**; erfolgen.

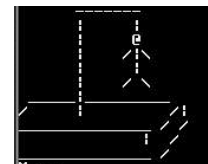
Experten-Aufgabe 3: Hangman (Profi-Aufgabe)



Erstellen Sie das Spiel Hangman!

Der Benutzer muss ein Wort erraten, das der Computer zufällig vorgibt.

Der Spieler nennt nun in beliebiger Reihenfolge nacheinander einzelne Buchstaben des Alphabets. Der PC gibt anschließend den erratenen Buchstaben an der richtigen Stelle des Wortes aus. So ergibt sich nach und nach das gesuchte Lösungswort. Kommt der Buchstabe darin jedoch nicht vor, so beginnt sich der Galgen über Teilstriche zu bilden. Ist das Bild des Galgen mit dem Männchen fertig, bevor der Spieler das Wort erraten hat, so ist das Spiel verloren.



Erweiterung: Entwickeln Sie das Program, so, dass unterschiedliche Schwierigkeitsgrade ausgewählt werden können!



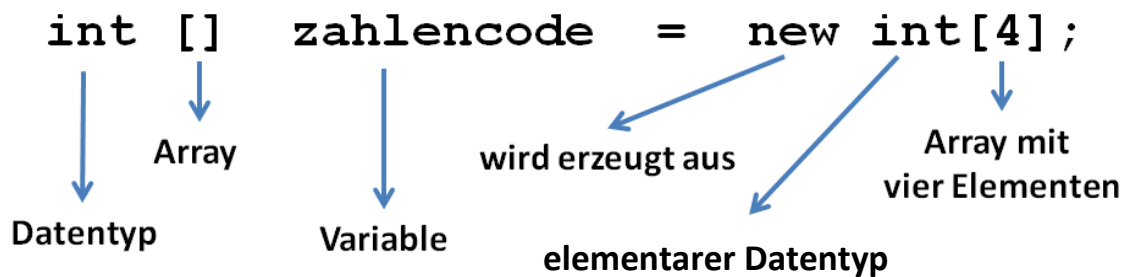
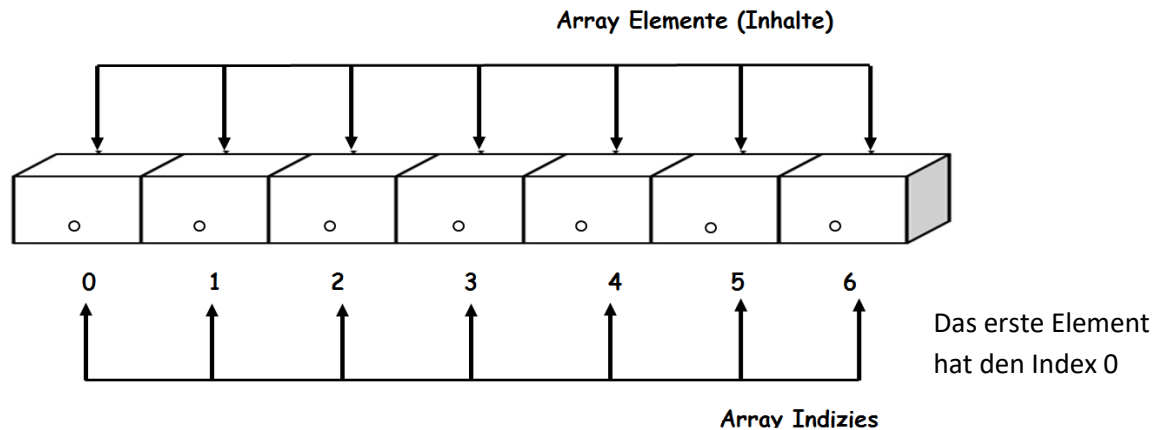
Bitte beginnen Sie nach dieser Aufgabe noch nicht mit den Arrays an! Sollten Sie mit der Hangman Aufgabe fertig sein, stellen Sie den Entwurf erst Ihrer Lehrkraft vor!

3. Arrays

Definition von Arrays

Ein Array (**Datenfeld**) ist eine Anordnung mehrerer Daten des gleichen Datentyps. Der Zugriff auf ein bestimmtes Element des Arrays erfolgt über den Array-Namen und einen Index, der in eckigen Klammern steht.

Ein eindimensionales Array kann als einzelige Tabelle (Schubladen) verstanden werden.



Beispielcode

...

```
int [] zahlencode = new int [4];
zahlencode [0] = 1;
zahlencode [1] = 4;
zahlencode [2] = 8;
zahlencode [3] = 6;
System.out.println("2. Zahl lautet " + zahlencode [1]); //Ausgabe von 4
```

Die Deklaration und die Initialisierung können auch **in einem Schritt** geschehen.

```
int [] zahlencode = {1,4,8,6};
```


Aufgabenstellung 1



Sheldon Cooper hat in seinem Schrank die Socken durcheinander gebracht. Er möchte die Socken in eine Kommode nach Tagen (Montag – Sonntag) einordnen. Jeden Tag trägt er eine andere Farbe.

Vervollständigen Sie das Programm, damit Sheldon jeden Tag neue Socken hat. Weitere Socken von Sheldon sind in den Farben *pink*, *schwarz* und *weiß*.

```
class Schrank
{
    public static void main(String[] args)
    {
        // Deklaration und Initialisierung
```



```
schrank[0] = "rote Socken";
schrank[1] = "gruene Socken";
schrank[2] = "blaue Socken";
schrank[3] = "graue Socken";
```

200104.docx



fisi_

```
System.out.println("Am Montag werden " + schrank [0] + " getragen");
System.out.println("Am Dienstag werden " + schrank [1] + " getragen");
System.out.println("Am Mittwoch werden " + schrank [2] + " getragen");
System.out.println("Am Donnerstag werden" + schrank [3] + " getragen");
System.out.println("Am Freitag werden " + schrank [4] + " getragen");
```



```
}
}
```

Aufgabenstellung 2



Sheldon hat bereits ein etwas komplizierteres Sockenprogramm geschrieben. An welchem Tag hat Sheldon welche Socken an? Überlegen Sie sich die Ergebnisse, ohne das Programm abzuschreiben.

```
class Sockenfarbe {
public static void main(String[] args) {
    // Deklaration und Initialisierung

    int [] tag = new int[7];

    tag [0] = 1;
    tag [1] = 3;
    tag [2] = 0;
    tag [3] = 4;
    tag [4] = 5;
    tag [5] = 2;
    tag [6] = 6;

    String []socken = new String[7];
    socken [0] = "rot";
    socken [1] = "blau";
    socken [2] = "gruen";
    socken [3] = "schwarz";
    socken [4] = "grau";
    socken [5] = "weiss";
    socken [6] = "pink";

    int y, ref;

    y = 0;
    ref = tag [y];
    System.out.print("Sockenfarbe am Montag ist ");
    System.out.println(socken [ref]);

    y = 1;
    ref = tag [y];
    System.out.print("Sockenfarbe am Dienstag ist ");
    System.out.println(socken [ref]);

    y = 2;
    ref = tag [y];
    System.out.print("Sockenfarbe am Mittwoch ist ");
    System.out.println(socken [ref]);

    y = 3;
    ref = tag [y];
    System.out.print("Sockenfarbe am Donnerstag ist ");
    System.out.println(socken [ref]);

    y = 4;
    ref = tag [y];
    System.out.print("Sockenfarbe am Freitag ist ");
    System.out.println(socken [ref]);

    y = 5;
    ref = tag [y];
    System.out.print("Sockenfarbe am Samstag ist ");
    System.out.println(socken [ref]);

    y = 6;
    ref = tag [y];
    System.out.print("Sockenfarbe am Sonntag ist ");
    System.out.println(socken [ref]);
}
```

Ausgabe: _____

Ausgabe: _____

Ausgabe: _____

Ausgabe: _____

Ausgabe: _____

Ausgabe: _____

Ausgabe: _____

Programmierauftrag I



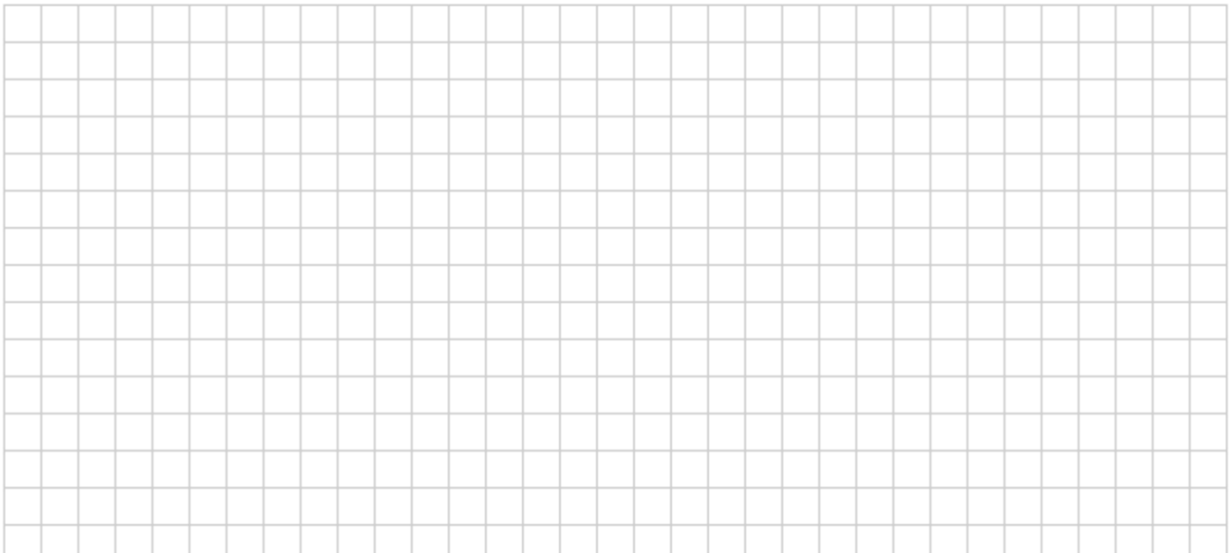
Sheldon möchte jetzt ein Programm, dass ihm nach Eingabe des gewünschten Tages als Zahl (Montag (1) – Sonntag (7)) die dazu vorgesehen Sockenfarbe ausgibt.

Erstellen Sie ein Struktogramm und einen Programmablaufplan. Entwickeln Sie anschließend den entsprechenden Java-Programmcode. Achten Sie auf eine strukturierte Darstellung des Programms und dokumentieren Sie wichtige Inhalte mit der Kommentarfunktion. Vergleichen Sie anschließend Ihren Programmcode mit Ihren Diagrammen!

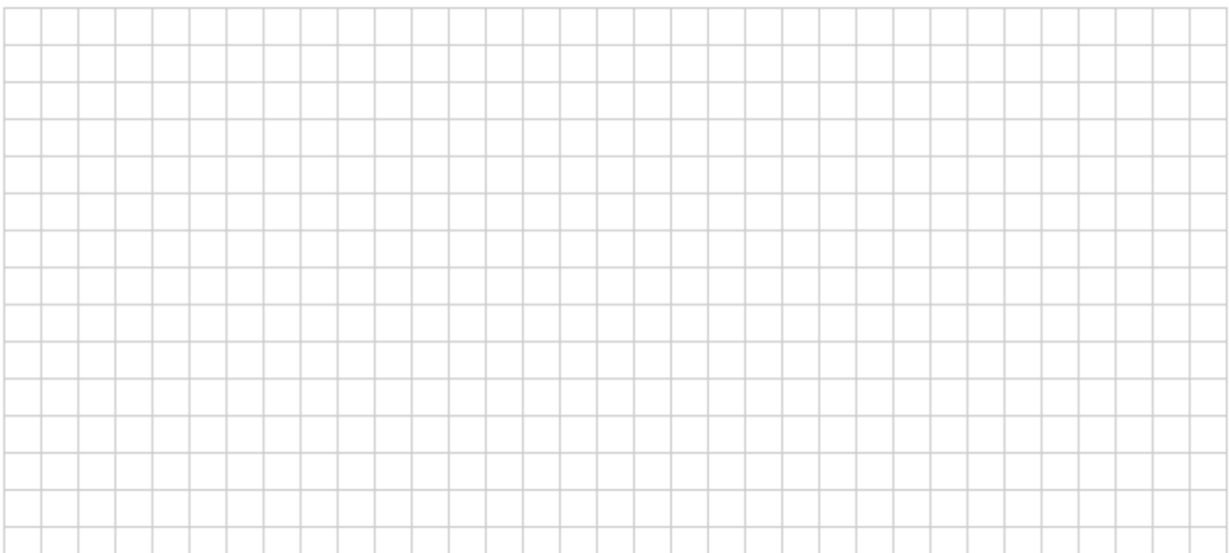


Für Experten: Erweitern Sie Ihr Programm. Geben Sie alle Sockenfarben aus mithilfe einer Schleife.

Struktogramm:

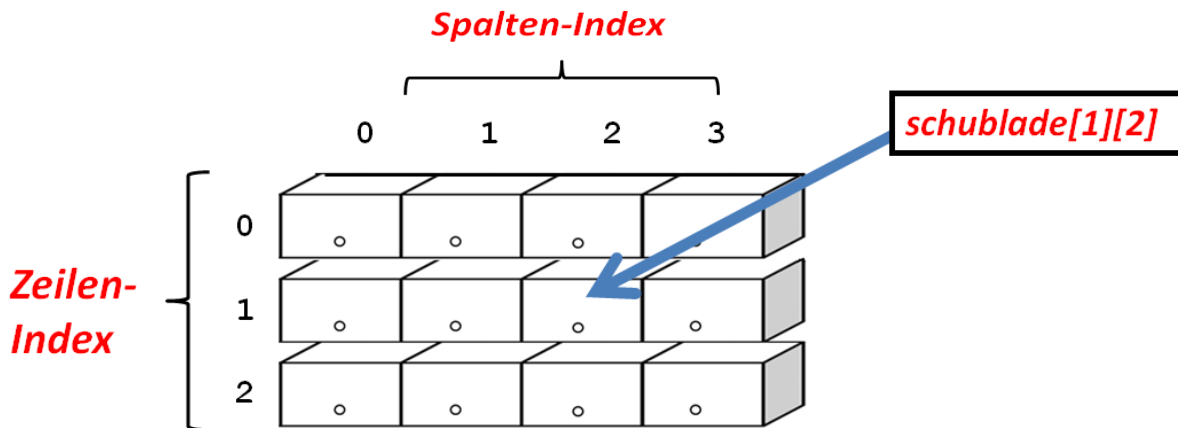


PAP:



Mehrdimensionale Arrays

Ein Array, dessen Elemente jeweils wieder Arrays sind, bezeichnet man als mehrdimensionales Array. Der Index beginnt für die Zeile und Spalte mit 0.



Deklaration & Initialisierung

```
Datentyp [][] arrayName = new Klasse [Zeilenanzahl][Spaltenanzahl];
String [][] schublade = new String[3][4];
```

! Bei der Initialisierung wird die **Anzahl** der Zeilen und Spalten angegeben. Bei drei Spalten nimmt der Index Werte von 0 bis 2 an.

Beispiel

```
String [][]smartphone = new String [2][3];
// Erste Reihe mit Smartphones
smartphone [0][0] = "HTC One";
smartphone [0][1] = "Apple iPhone 6";
smartphone [0][2] = "Samsung Galaxy S5";

// Zweite Reihe mit Smartphones
smartphone [1][0] = "Sony Xperia Z3";
smartphone [1][1] = "Nokia Lumia 535";
smartphone [1][2] = "Samsung Galaxy Alpha";
```

Deklaration & Initialisierung in einem Schritt

```
int[][] cartCoords = { {0, 0}, {2, 1}, {3, 1}, {-1, -2} };
```

Die Länge der einzelnen Unter-Arrays ist beliebig, solange sie denselben Elementtyp haben. Im folgenden Beispiel wird diese Möglichkeit dazu genutzt, ein pascalsches Dreieck¹ darzustellen:

```
int[][] pascal = { {1}, {1, 1}, {1, 2, 1} };
```

```

      1
     1 1
    1 2 1
```

¹ https://de.wikipedia.org/wiki/Pascalsches_Dreieck

Aufgabenstellung

Ein Baumarkt verwendet zur Verwaltung nachfolgender Produkte ein mehrdimensionales Array. Die Deklaration und Initialisierung wurde bereits wie in der Tabelle realisiert.

<i>Lampe</i>	<i>Schraubenzieher</i>	<i>Akkuschrauber</i>	<i>Säge</i>
<i>Brechstange</i>	<i>Zange</i>	<i>Hammer</i>	<i>Mehrfachstecker</i>

Anschließend wurden die Inhalte mithilfe des folgenden Programmcodes im Lager umgeräumt.

```
baumarkt [0][0]= baumarkt [1][1];
baumarkt [0][1]= "Akkuschrauber";
baumarkt [0][2]= baumarkt [1][2];
baumarkt [1][0]= baumarkt [0][3];
baumarkt [1][1]= baumarkt [0][2];
```

fisi_Schleifen_Vk_20200104.docx



Welche Ausgaben liefert der folgende Programmcode?

Kontrollieren Sie Ihre Ergebnisse in der Gruppe mithilfe des Lösungswortes!

*Gesuch-
ter Wert*

*Lösungs-
Buchstabe*

```
System.out.println(baumarkt [0][0]);
```

Lampe

K

Akkuschrauber

Z

Zange

A

```
System.out.println(baumarkt [0][1]);
```

Brechstange

o

Akkuschrauber

r

Schraubenzieher

a

```
System.out.println(baumarkt [0][2]);
```

Hammer

r

Akkuschrauber

t

Zange

a

```
System.out.println(baumarkt [1][0]);
```

Brechstange

z

Zange

r

```
System.out.println(baumarkt [1][1]);
```

Säge

a

Brechstange

e

Hammer

y

Akkuschrauber

o

Lösungswort: _____

Programmierauftrag II: Mit Amazon durchs Chaos-Lager

Sheldon hasst Chaos und Unordnung. Er ist zutiefst betrübt als er erfährt, dass bei Amazon chaotisch gelagert wird und das auch noch mit voller Absicht.

Hinter der scheinbaren Unordnung steckt jedoch ein System: die chaotische Lagerung. Das chaotische Lager ist eine Ansammlung von Regalen, das Platz für die Produkte bietet.



Einen Ausschnitt dieses chaotischen Systems (nächste Seite) will Sheldon nun selbst computer-gestützt entwickeln.

- Erfassen Sie dazu die unten gezeigten Produkte in einem mehrdimensionalen Array
- Nach Eingabe der entsprechenden Zeile (1-3) und Spalte (1-4) soll das zugehörige Gerät ausgegeben werden.

Erweiterung: Falls Produkte aus der ersten Zeile gewählt werden, soll die Meldung „Produkte im Angebot“ ausgegeben werden.

Erstellen Sie ein Java-Programm zur Lösung der genannten Problemstellung!



Ausschnitt des Chaos-Lagers

Programmierauftrag III: Weitere Aufgaben im Chaos Lager

Erstellen Sie die folgenden Aufgaben jeweils in einer neuen Java-Datei.

Aufgabe E1

Geben Sie am Ende den gesamten Lagerinhalt aus.

Verwenden Sie eine entsprechende Schleife!

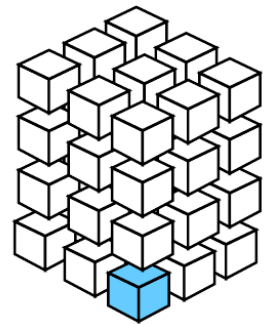
Aufgabe E2

Der Anwender soll ein bestimmtes Produkt eingeben und anschließend soll ihm der Lagerort (Zeile, Spalte) ausgegeben werden. Füllen Sie davor das gesamte Lager mit Inhalten!

Aufgabe E3

Das Lager von Amazon soll jetzt dreidimensional aufgeteilt werden.

Erweitern Sie das bisher zweidimensionale Lager um eine weitere Dimension und ergänzen Sie es um selbst gewählte Artikel. Im Anschluss sollen alle Artikel ausgegeben werden. Nach Eingabe des gewünschten Produktes soll der genaue Lagerort (x-, y- und z- Koordinate) ausgegeben werden.



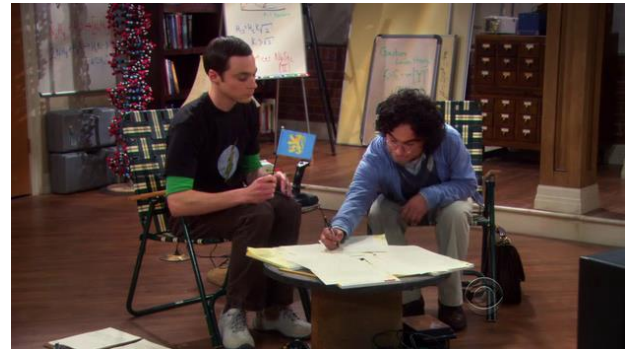
Wie kann ein dreidimensionales Array deklariert und initialisiert werden?

Informieren Sie sich selbstständig über die benötigten Inhalte.

Raum für Notizen:

The Roommate Agreement

Sheldon hat eine schlimme Vermutung. Sein Mitbewohner Leonard hält sich nicht an die Mitbewohnervereinbarung (roommate agreement) und verändert heimlich am Thermostat die Zimmertemperatur in ihrer gemeinsamen Wohnung. Ein Auszug der Mitbewohnervereinbarung über die vertraglich vereinbarte Temperatur ist unten aufgeführt.



Sheldon hat nun über die letzten zwei Wochen Messungen der Zimmertemperatur vorgenommen. Die Ergebnisse hat er in folgender Tabelle zusammengefasst.

Tag	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Temperatur in °C	22.5	22.4	22.3	22.1	21.9	21.8	21.5	21.4	21.6	21.5	21.5	21.6	21.5	22.5

The Roommate Agreement

- Die durchschnittliche Zimmertemperatur muss 22°C betragen mit einer maximal erlaubten Abweichung von $\pm 0.5^\circ\text{C}$ über zwei Wochen.
- Die Tagestemperatur darf die tatsächliche durchschnittliche Zimmertemperatur maximal um 0.5°C über- bzw. unterschreiten.
- Die maximale Temperaturabweichung auf zwei aufeinanderfolgenden Tagen darf maximal 0.7°C betragen.

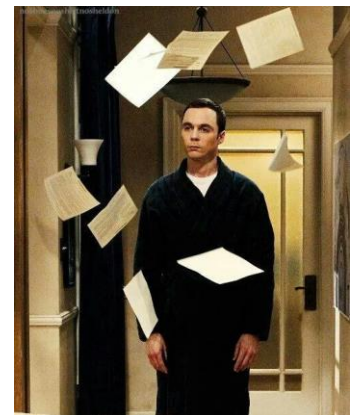


Helfen Sie Sheldon und entwickeln Sie ein Java-Programm das die vertraglichen Bestandteile des Roommate Agreements überprüft! Welche Regeln wurden an welchen Tagen nicht eingehalten?

Achten Sie auf eine strukturierte Darstellung des Programmes und dokumentieren Sie wichtige Inhalte mithilfe der Kommentarfunktion!

Für Profis:

- Wie viele Verstöße finden sich insgesamt nach der Mitbewohnervereinbarung?
- Wie hoch ist die maximale und minimale Temperatur?



Die Reaktion von Sheldon auf eine Verletzung der Mitbewohnervereinbarung finden Sie unter dem folgenden Link: <https://www.youtube.com/watch?v=HpeF067Hx2Y>



Aufgaben für absolute Experten

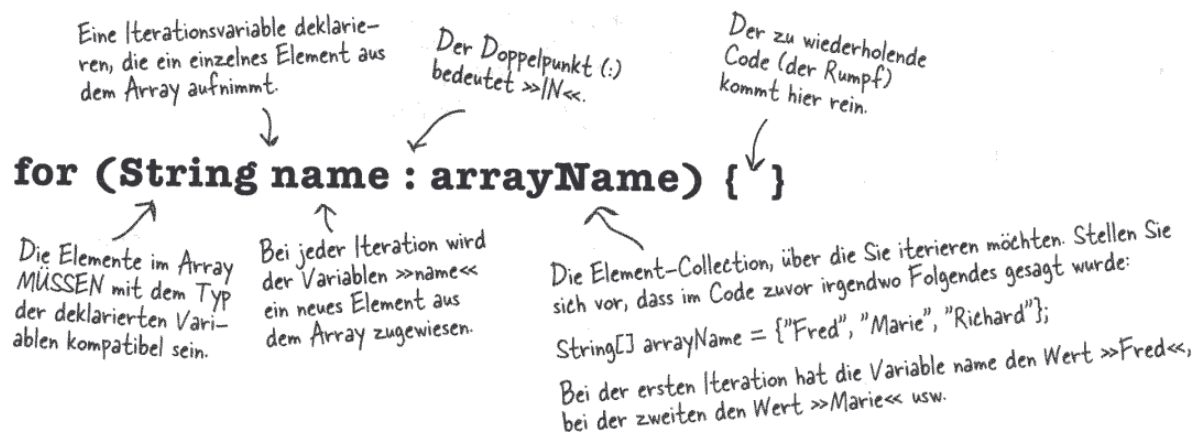


ForEach-Schleife

Java besitzt neben der bisher bekannten For-Schleife noch eine zweite For-Schleife.

Die ForEach-Schleife ist ein einfaches Mittel um verschiedene Elemente zu durchlaufen. ForEach Schleifen besitzen keine nach außen sichtbare Laufvariable, sondern geben immer direkt den Wert zurück.

Aufbau



Ablauf der ForEach-Schleife

- Es wird eine String Variable namens *name* erzeugt.
- Name wird der erste Wert in *arrayName* zugeordnet
- Der Schleifenkörper (Codeblock in den geschweiften Klammern) wird ausgeführt
- *name* wird der nächste Wert in *arrayName* zugeordnet
- Der Vorgang wird sooft wiederholt, bis alle Elemente durchlaufen sind

Beispiel

```
// Ein Array mit 9 Werten wird erzeugt
int[] array = {4, 8, 4, 2, 2, 1, 1, 5, 9};

// ForEach Schleife
for( int k: array ) {
    System.out.println("k = "+k);
}
```

Die Ausgabe ist:

```
k = 4
k = 8
k = 4
k = 2
k = 2
k = 1
k = 1
k = 5
k = 9
```

Aufgabenstellung

Sie entdecken auf einem Serverlaufwerk eine seltsame Textdatei mit einer Art Zahlencode.

Sie vermuten, dass diese Zahlen vom Typ *int* (*Integer*) in den Typ *char* umgewandelt werden müssen.

Allgemein erfolgt eine **Typumwandlung (cast)** mit:

```
charWert = (char)intWert;
```

Erstellen Sie ein Programm, das den Zahlencode entziffert.

Die Botschaft wurde bereits in einem Array unter `R:\austausch\lehrer\martin.voelkl\AP\Botschaft.txt` gespeichert.



Die geheime Botschaft:

Hinweis: Text
ist US-ASCII 7bit

```
int [] intWerte =
{73,104,114,95,115,101,105,100,95,100,105,101,95,66,101,115,116,101,110,95,7
2,111,108,116,95,101,117,99,104,95,101,117,114,101,95,71,101,104,97,108,116,
115,101,114,104,111,101,104,117,110,103,95,118,111,114,110,101,95,97,98,33}
```

Foobar Aufgabe (optional)

Diese Aufgabe wird in der IT oft in Vorstellungsgesprächen und Einstellungstest von den Teilnehmerinnen und Teilnehmern auf dem Blatt Papier erwartet. Versuchen auch Sie die Aufgabe zuerst ohne Rechner zuerst zu lösen!

Schreibe ein Programm, das die Zahlen 1 bis 100 (jeweils inklusive) auf einer eigenen Zeile ausgibt. Dabei soll jede durch 3 teilbare Zahl durch die Ausgabe "foo", jede durch 5 teilbare Zahl durch "bar" und jede durch 3 und 5 teilbare Zahl durch "foobar" ersetzt werden.

Die ersten Ausgaben des Programms sähen also so aus:

```
1
2
foo
4
bar
foo
7
8
foo
bar
...
```



Aufgabe für absolute Experten (optional)

Falls Sie jetzt schon fertig sind, können Sie hier eine Abschlussprüfungsaufgabe aus der Fachqualifikation für Anwendungsentwickler bearbeiten. Die Aufgabe ist nur für erfahrene SchülerInnen gedacht und muss in der Abschlussprüfung handschriftlich gelöst werden.



Die IT-Systemprofi GmbH soll für den Webshop Anbieter „Digital Saloon“ eine Funktion entwickeln, die für Kreditkartennummern eine Prüfziffernberechnung nach dem Luhn-Algorithmus durchführt.

Beispiel für die Prüfziffernberechnung der Kreditkartennummer 2718281828458567 nach dem Luhn-Algorithmus:

	Ziffernstelle																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	PZ	Ergebnis
Kreditkarten-	2	7	1	8	2	8	1	8	2	8	4	5	8	5	6	7	
Schritt 1	4		2		4		2		4		8		16		12		
Schritt 2	4		2		4		2		4		8		1+6		1+2		34
Schritt 3		7		8		8		8		8		5		5			49
Schritt 4	34 + 49																83
Schritt 5	83 auf nächstgrößere durch 10 teilbare Zahl aufrunden																90
Schritt 6	90-83																7(PZ*)

* PZ = Prüfziffer

- Schritt 1: Multiplikation aller Ziffern an ungerader Stelle mit 2
- Schritt 2: Bildung der Quersummen aller entstandenen Produkte und Addition aller entstandenen Quersummen
- Schritt 3: Addition aller Ziffern an gerader Stelle
- Schritt 4: Addition der Ergebnisse aus den Schritten 2 und 3
- Schritt 5: Aufrundung des Ergebnisses aus Schritt 4 auf die nächstgrößere durch 10 teilbare Zahl
- Schritt 6: Berechnung der Differenz aus dem Ergebnis aus Schritt 5 und dem Ergebnis aus Schritt 4

Die Kreditkartennummer wird der Funktion als String übergeben. Stimmen die letzte Ziffer der Kreditkartennummer und die errechnete Zahl überein, ist die Kreditkartennummer in Ordnung. In diesem Fall gibt die Funktion true, sonst false zurück.

Stellen Sie eine entsprechende Funktion in Pseudocode in einem Struktogramm nach DIN 660261 oder PAP nach DIN 66001 dar!