



# JAVA-Workshop

## Workshop4Schools

Martin Bratoi, Sebastian Menzel  
Dienstag, 01.08.2023

PUBLIC

THE BEST RUN



# Agenda

- Was ist Java?
- Programmstruktur
- Variablen
- Operatoren
- Kontrollstrukturen
- Methoden
- Eingabe & Ausgabe

# Was ist Java?



# Motivation

- Ziele von Programmen:
  - Das Leben erleichtern
  - Probleme vereinfachen
  - Menschen unterstützen
- Steigende Relevanz im digitalen Zeitalter:
  - In vielen Berufszweigen gefragt / gefordert
  - Immer mehr Fachkräftebedarf
  - Gute Bezahlung

**Warum gibt es nicht nur eine  
Programmiersprache?**

# Warum Java?

- Geschichte:
  - Entwickelt von Sun Microsystems (Gehört heute zu Oracle)
  - März 1995 erstmals vorgestellt
  - Wurde zur verbreitetsten objektorientierten Sprache
- Vorteil: Betriebssystemunabhängig
- Verwendungszweck:
  - Webanwendungen (Browseranwendungen mit Server- und Clientteilen)
  - Desktopanwendungen (z.B. Minecraft)
  - Apps (Android)

# Programmstruktur (Syntax)

# Befehl

- Atomare Einheit im Programm
- Dient der Erreichung des Programmzwecks
- Wird mit einem Semikolon beendet



```
System.out.print("Hello World");
```

Gibt einen Wert in der Konsole aus

→ "Hello World"



```
System.out.println("Hello World");
```

Gibt einen Wert in der Konsole aus und beginnt eine neue Zeile

→ "Hello World [Zeilenumbruch]"



# Anweisungsblock


- Führt mehrere Befehle hintereinander aus
- Geschweifte Klammern
- Für jeden Befehl eine eigene Zeile



```
{  
    System.out.println( "Hello World" );  
    System.out.println( "Good Morning" );  
}
```

# Klassen

- Programme können groß und unüberschaubar werden
- Bessere Strukturierung



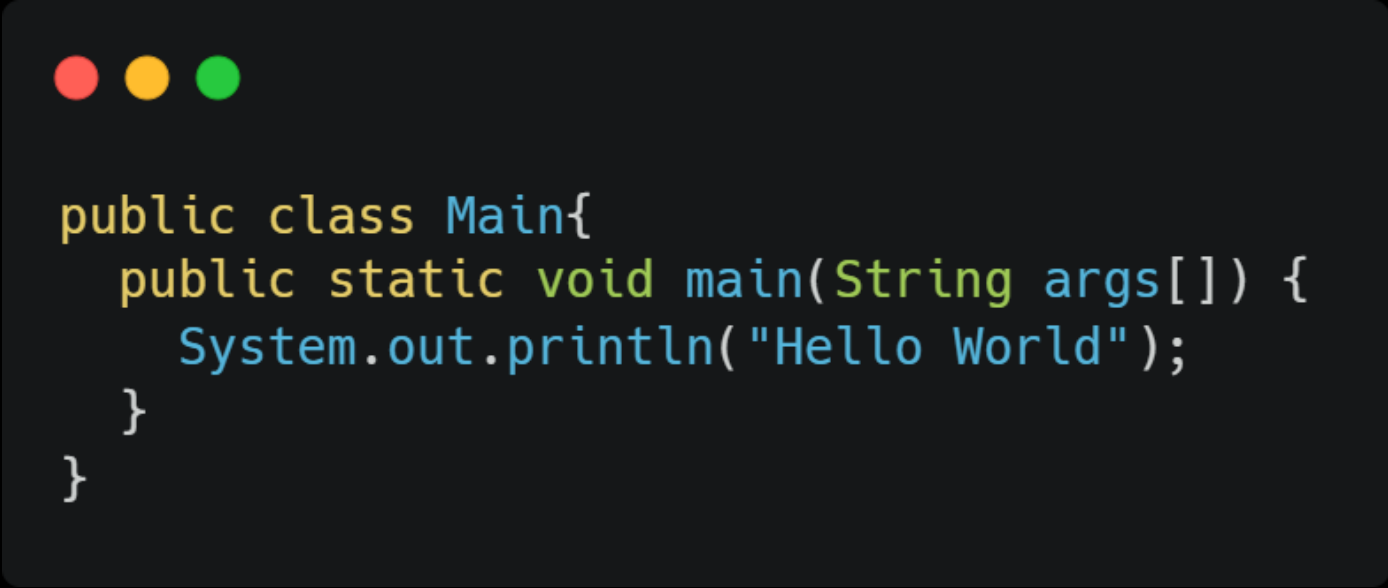
```
public class Main{  
    public static void main(String args[]) {  
        System.out.println("Hello World");  
        System.out.println("Good Morning");  
    }  
}
```

# Entwicklungsumgebung Eclipse



# Übung: Hello World

1. Schreibe ein Java-Programm, welches „Hello World!“ in der Konsole ausgibt!



```
public class Main{  
    public static void main(String args[]) {  
        System.out.println("Hello World");  
    }  
}
```

Die Klassendefinition und main-Methode werden bei weiteren Lösungen weggelassen!

2. Modifiziere das Programm, sodass es deinen Namen oder einen beliebigen Text ausgibt!

# Variablen



# Variablen in der Mathematik

Lineare Gleichungssysteme:

$$x = 4$$

$$y = 11 - 2 * x$$

Wie lösen wir das?

Einsetzen:  $y = 11 - 2 * 4$

# Variablen in der Informatik



```
int x;
```


```
x = 4
```

```
y = 11 - 2 * x
```

```
System.out.println(11 - 2 * x); // 3
```

→ Wir können in unseren Programmen  
Platzhalter (Variablen) definieren!

# Ein praktisches Beispiel: Partyeinladungen verschicken



```
class Main {  
    public static void main(String args[]) {  
        String name = "Leon";  
        System.out.print("Hallo ");  
        System.out.print(name);  
        System.out.print(", ich feiere am Samstag um 20 Uhr in der Gartenstraße. ");  
        System.out.print("Bitte bring Snacks mit. :)");  
    }  
}
```



## Übung: Variablen

Du möchtest nicht für jede Geburtstagsfeier eine neue Einladung schreiben. Schreibe dein Programm so, dass du nur einmal das Alter ändern musst. Auch soll die Einladung wie vorher gezeigt für alle Freunde gelten.

### Beispielausgabe:

Hallo Leon, ich feiere am Samstag um 20 Uhr meinen 17. Geburtstag. Bitte bring Snacks mit. :)

Hallo Lea, ich feiere am Samstag um 20 Uhr meinen 18. Geburtstag. Bitte bring Snacks mit. :)

Tipp:

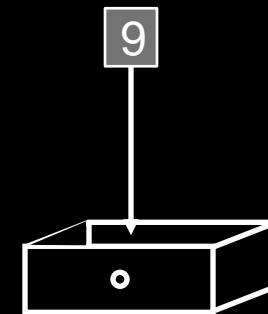
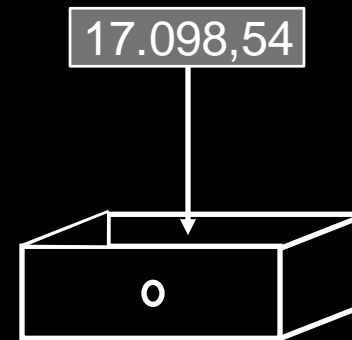
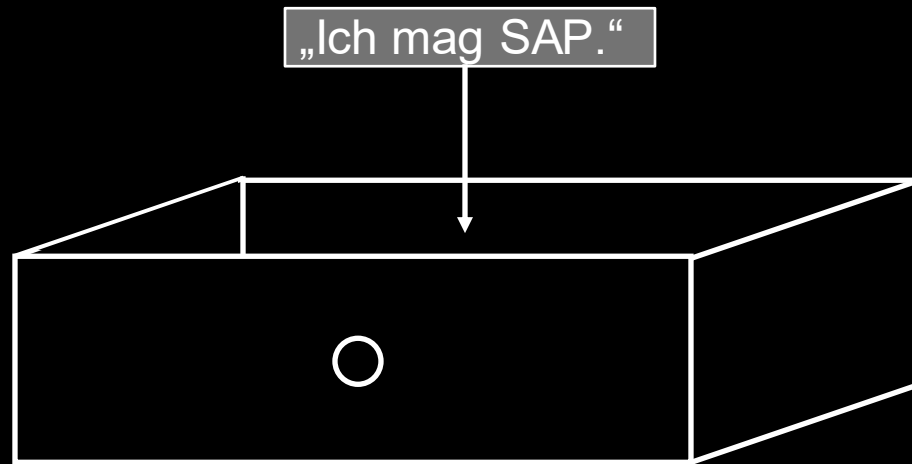
Benutze eine Variable für den Namen und für das Alter.  
Benutze mehrere "System.out.print()" Anweisungen.



# Was ist jetzt dieses 'int'?

Analogie aus der Küche: Man hat viele verschiedene Schubladen. In die Kleinen kommt das Besteck, in die Großen die Töpfe.

So läuft das auch in Java:



# Beispiele für Datentypen



```
String name = "Kim";  
char anfangsbuchstabe = 'K';
```

```
double gehalt = 1.250;  
float groesse = 1.72f;
```

```
int plz = 67098;  
short schlafProNacht = 9;  
byte freunde = 2;
```

# Warum brauchen wir verschiedene Datentypen?

Datentyp	Größe	Beschreibung
byte	1 byte	Speichert ganze Zahlen von -128 to 127
short	2 bytes	Speichert ganze Zahlen von -32.768 to 32.767
int	4 bytes	Speichert ganze Zahlen von -2.147.483.648 to 2.147.483.647
long	8 bytes	Speichert ganze Zahlen von -9.223.372.036.854.775.808 to 9.223.372.036.854.775.807
float	4 bytes	Speichert Dezimalzahlen mit 6-7 Nachkommastellen
double	8 bytes	Speichert Dezimalzahlen mit 15 Nachkommastellen
boolean	1 bit	Speichert Wahrheitswerte (true/false)
char	2 bytes	Speichert ein einzelnes Zeichen (bspw. Buchstaben)

# Übung: Welcher Datentyp passt?

- |                         |               |
|-------------------------|---------------|
| • Klassenstufe          | byte          |
| ■ Notenschnitt          | float         |
| ■ Ersparnisse           | double, float |
| ■ Haustierrasse         | String        |
| ■ Taschengeld pro Monat | float, double |
| ■ Anfangsbuchstabe      | char          |

## **Tipp:**

Manchmal gibt es mehr als eine Lösung.

# Was ist bei Variablen zu beachten?

Bezeichner klein schreiben.

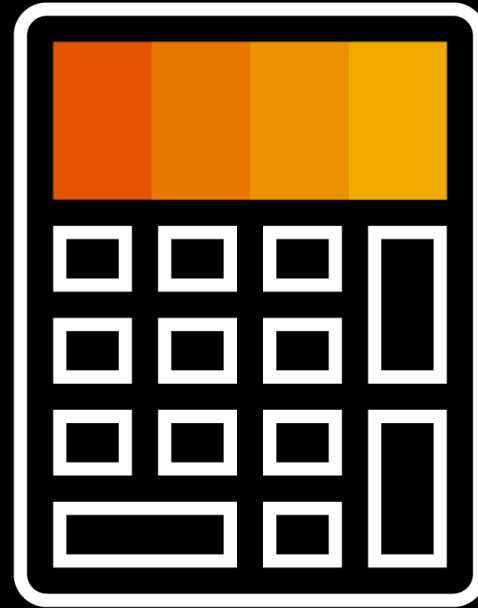
String groß schreiben.

Datentyp angeben.

```
String name = "Linus Peterson";  
String strasse = "Rheinfeldstrasse";  
int hausnummer = 17;  
int plz = 67063;  
String stadt = "Ludwigshafen";
```

"" um den Inhalt von Strings.

# Operatoren



# Zuweisung

```
int x = 3;  
String y = "Hello";  
int z = x;
```



weist dem linken Operanden den Wert des rechten Operanden zu

# Rechenoperatoren

## Addition:



```
int x = 2 + 9; //11
String y = "Hello " + "World!"; //Hello World!
```

## Subtraktion:



```
int x = 4 - 2; //2
double y = 9.7 - 4.7; //5
```

## Multiplikation:



```
int x = 3 * 3; //9
double y = 4.5 * 2.5; //11.25
```

## Division:



```
int x = 9 / 3; //3
double y = 1 / 2; //0.5
int z = 1 / 2; //0
```

## Modulo:



```
int x = 11 % 3; //2
int y = 8 % 4; //0
```



# Kurzschreibweisen

```
x++;
```



```
x = x + 1;
```

```
x--;
```



```
x = x - 1;
```

# Übung: Operatoren

1. Erstelle die Variable `strecke` (`double`) und weise ihr den Wert `3.5` [km] zu.
2. Erstelle die Variable `minuten` (`int`) und weise ihr den Wert `20` zu.
3. Berechne, wie viele [km] du pro Minute zurückgelegt hast und speichere das in einer neuen Variable `kmProMinute` (`double`). Tipp: Nutze den Operator `/`, um `strecke` durch `minuten` zu teilen.
4. Dir gefällt die Einheit [km/m] nicht und möchtest das in [km/h] umrechnen. Speichere dein Ergebnis in einer neuen Variable `kmProStunde` (`double`)
5. Gib den Wert für `kmProStunde` in der Konsole aus.

## Lösung: Operatoren



```
public static void main(String[] args){  
  
    double strecke = 3.5;  
    int minuten = 20;  
    double kmProMinute = strecke / minuten;  
    double kmProStunde = kmProMinute * 60;  
    System.out.println(kmProStunde);  
  
}
```



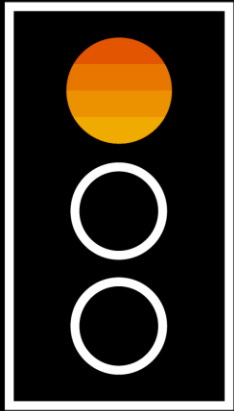
# Kontrollstrukturen

## Steuerung des Programmflusses

# Bedingte Anweisung



# Real-Life Beispiele



**Wenn** die Ampel grün ist, **dann** darf ich über die Straße.  
**Ansonsten** muss ich warten.

**Wenn** Kaffee leer, **dann** neuen holen.  
**Ansonsten** trinken!



# Bedingte Anweisung in JAVA

```
if(<Bedingung>) {  
    <Anweisung>  
} else {  
    <Anweisung>  
}
```

“Wenn...”

“...dann...”

“...sonst...”

**else** ist optional!

# Bedingungen

Wahr  oder Falsch 

```
boolean bedingung = true;  
  
if(bedingung) {  
    System.out.println("Das wird ausgegeben!");  
}
```

```
boolean bedingung = false;  
  
if(bedingung) {  
    System.out.println("Das wird nicht ausgegeben!");  
}
```



# Vergleichsoperatoren

## Gleichheit (==):

```
5 == 5 //true  
5 == 3 //false
```

## Größer (>)

```
3 > 5 //false  
9 > 2 //true
```

## Größer-Gleich (>=):

```
9 >= 3 //true  
5 >= 5 //true
```

## Ungleichheit (!=):

```
7 != 5 //true  
3 != 3 //false
```

## Kleiner (<):

```
3 < 5 //true  
9 < 2 //false
```

## Kleiner-Gleich (<=):

```
9 <= 3 //false  
5 <= 5 //true
```

## Quiz: Bedingte Anweisung



```
int x = 5;
int y = x + 3;

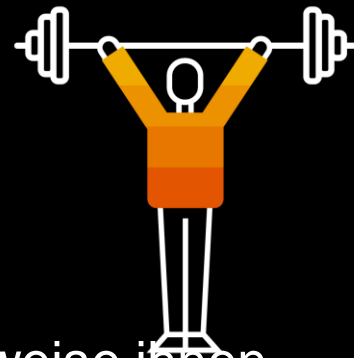
if(y != 7){
    System.out.println("Lösung A");
} else {
    System.out.println("Lösung B");
}
```

1) A oder B

```
int x = 10;

if(x < 5){
    System.out.println("Lösung A");
} else if (x < 10) {
    System.out.println("Lösung B");
} else {
    System.out.println("Lösung C");
}
```

2) A, B oder C



## Übung: Bedingte Anweisung

1. Erstelle zwei Variablen `alterMarie` und `alterMax` mit passenden Datentypen und weise ihnen beliebige positive Werte zu.
2. Prüfe, ob Marie und Max gleich alt sind. Wenn dies zutrifft, soll „gleich alt“ in der Konsole ausgegeben werden.
3. Füge weitere Bedingungen hinzu, um zu prüfen, welcher der beiden nun älter ist. Ist Marie älter, so wird „Marie ist älter“ ausgegeben, ansonsten „Max ist älter“.

**Expertenaufgabe:** Für die Umsetzung des Programmablaufs in Aufgabe 3 gibt es mehrere Lösungen. Finde mindestens eine weitere.

# Lösung: Bedingte Anweisung

Nach Aufgabe 3



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

    int alterMarie = 17;
    int alterMax = 18;
    if(alterMarie == alterMax){
        System.out.println("gleich alt");
    }else if(alterMarie > alterMax){
        System.out.println("Marie ist älter");
    }else{
        System.out.println("Max ist älter");
    }

}
```

Alternative



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

    int alterMarie = 17;
    int alterMax = 18;
    if(alterMarie == alterMax){
        System.out.println("gleich alt");
    }else{
        if(alterMarie > alterMax){
            System.out.println("Marie ist älter");
        }else{
            System.out.println("Max ist älter");
        }
    }

}
```

# Schleifen



# Real-Life Beispiele



**Solange** ich kein JAVA kann,  
muss ich üben!

Mache **10 mal** einen Korbleger.



# Schleifen in JAVA



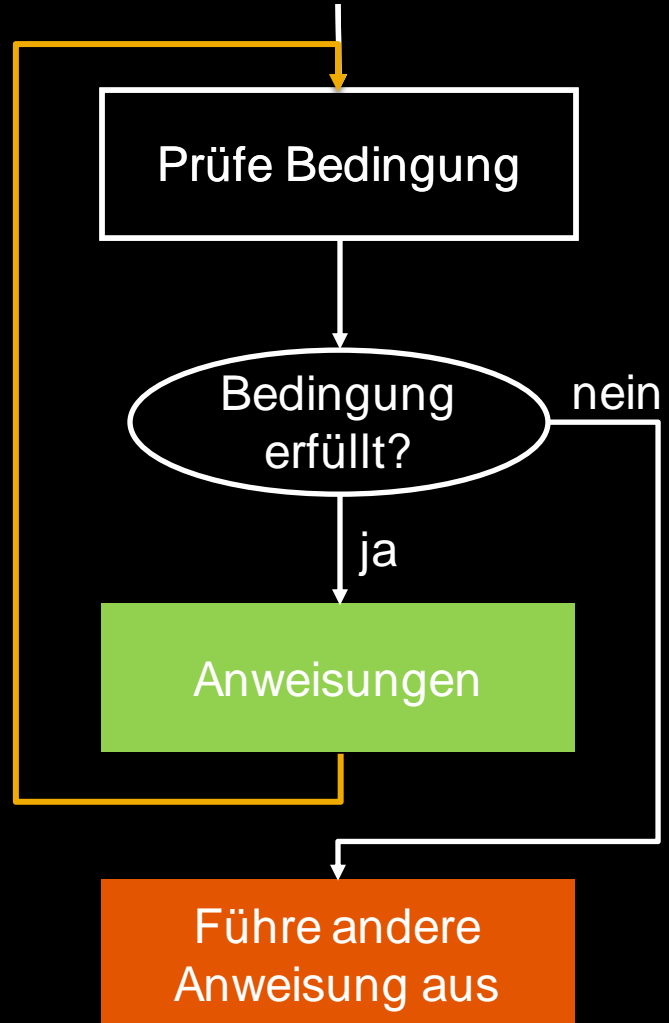
**while**  
(solange...)

**vs**



**for**  
(so oft...)

# while-Schleife in JAVA



⇒ `while(<Bedingung>){`  
    <Anweisung>  
}



# for-Schleife in JAVA

```
for(<Startwert>; <Bedingung>; <Zähler>) {  
    <Anweisung>  
}
```

# for-Schleife in JAVA

`for( <Startwert>`

Definition und Startwert  
der lokalen Laufvariablen

```
int i = 0
```

`;`

`<Bedingung>`

So lange wird  
durchgeführt (vgl. while)

```
i < 10
```

`;`

`<Zähler>`

Anpassung der Laufvariablen  
nach jeder Durchführung

```
i++
```

`)`

## Quiz: for-Schleife



```
for(int i = 0; i < 10; i++) {  
    System.out.println(i);  
}
```

1) Zahlen 1 bis 10

2) Zahlen 0 bis 10

3) Zahlen 0 bis 9

# Übung: Schleifen

Programmiere einen Raketen-Countdown, der von 10 auf 0 herunterzählt und dann die Rakete starten lässt...

1. ...mit einer `while`-Schleife
2. ...mit einer `for`-Schleife

Beispielausgabe:

10

9

8

7

6

5

4

3

2

1

Rakete startet...

# Lösung: Schleifen

## Aufgabe 1:

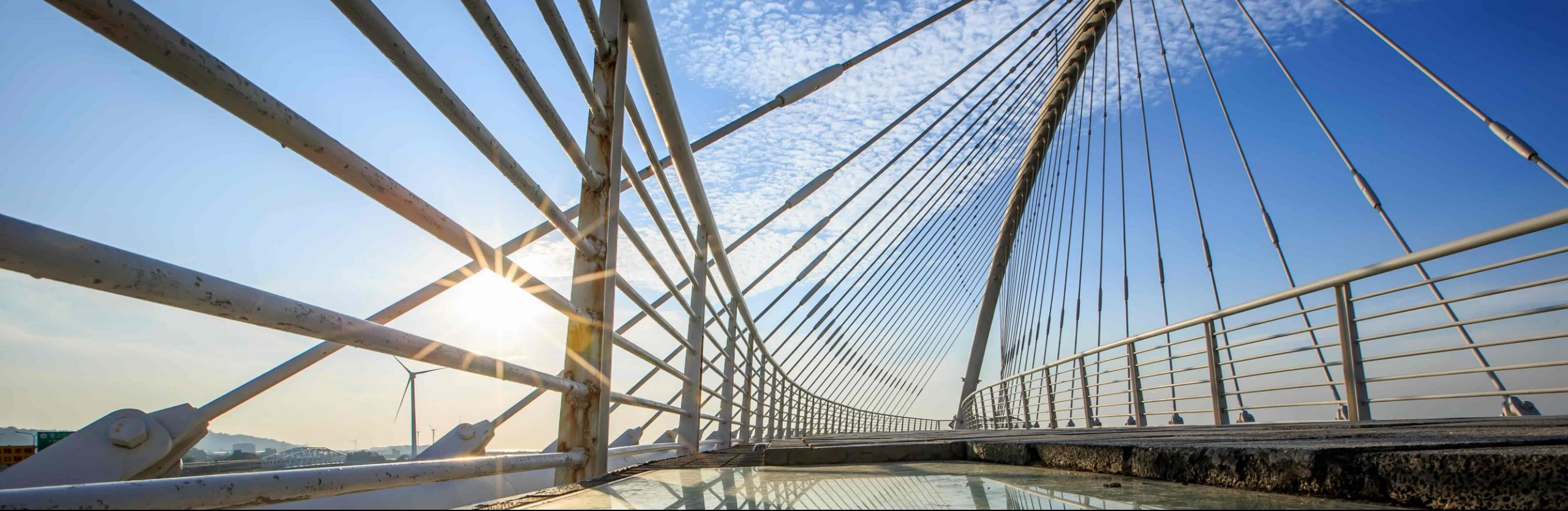
```
public static void main(String[] args) {  
    int i = 10;  
    while(i >= 0) {  
        System.out.println(i);  
        i--;  
    }  
}
```

## Aufgabe 2:

```
public static void main(String[] args) {  
    for(int i = 10; i >= 0; i--) {  
        System.out.println(i);  
    }  
}
```

**"Never code twice."**

**- Jeder Entwickler, immer**

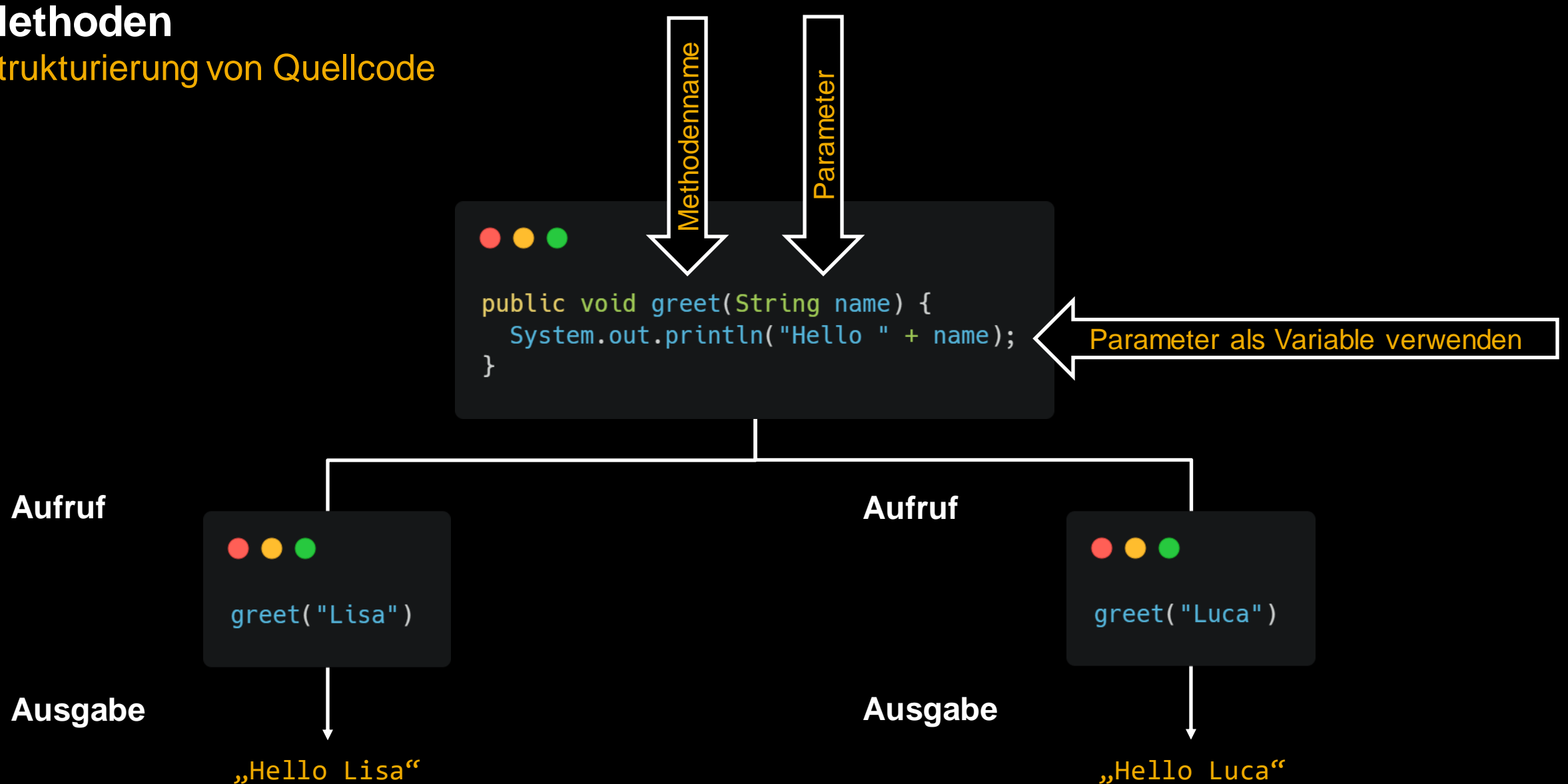


# Methoden

Strukturierung von Quellcode

# Methoden

## Strukturierung von Quellcode





# Übung: Methoden

1. Erstelle eine Methode `pruefeAlter`, die eine Zahl `alter` als Parameter entgegen nimmt. Prüfe, ob dieser Parameter größer gleich `18` ist. Trifft dies zu, so soll in der Konsole „`reinlassen`“ ausgegeben werden. Im anderen Fall soll „`wegschicken`“ in der Konsole erscheinen.
2. Rufe die Methode `pruefeAlter` in deiner Hauptmethode einmal mit dem Wert `22` und mit dem Wert `14` auf. Nach dem ersten Aufruf sollte „`reinlassen`“ und nach dem zweiten Aufruf „`wegschicken`“ angezeigt werden. Dann hast du alles richtig gemacht.
3. Erweitere die Methode `pruefeAlter` um einen zweiten Parameter `mindestAlter`. Wir brauchen diesen Parameter, da das Mindestalter nicht überall gleich ist. Ersetze bei der Altersüberprüfung die Zahl `18` durch den neuen Parameter `mindestAlter`. Setze das Mindestalter bei den Aufrufen auf `21`.

# Lösung: Methoden

Nach Aufgabe 1 & 2



```
public class Main{

    public void pruefeAlter(int alter) {
        if(alter >= 18){
            System.out.println("Volljährig");
        }else{
            System.out.println("Ausweiskontrolle");
        }
    }

    public static void main(String args[]) {
        pruefeAlter(22);
        pruefeAlter(14)
    }

}
```

Nach Aufgabe 3



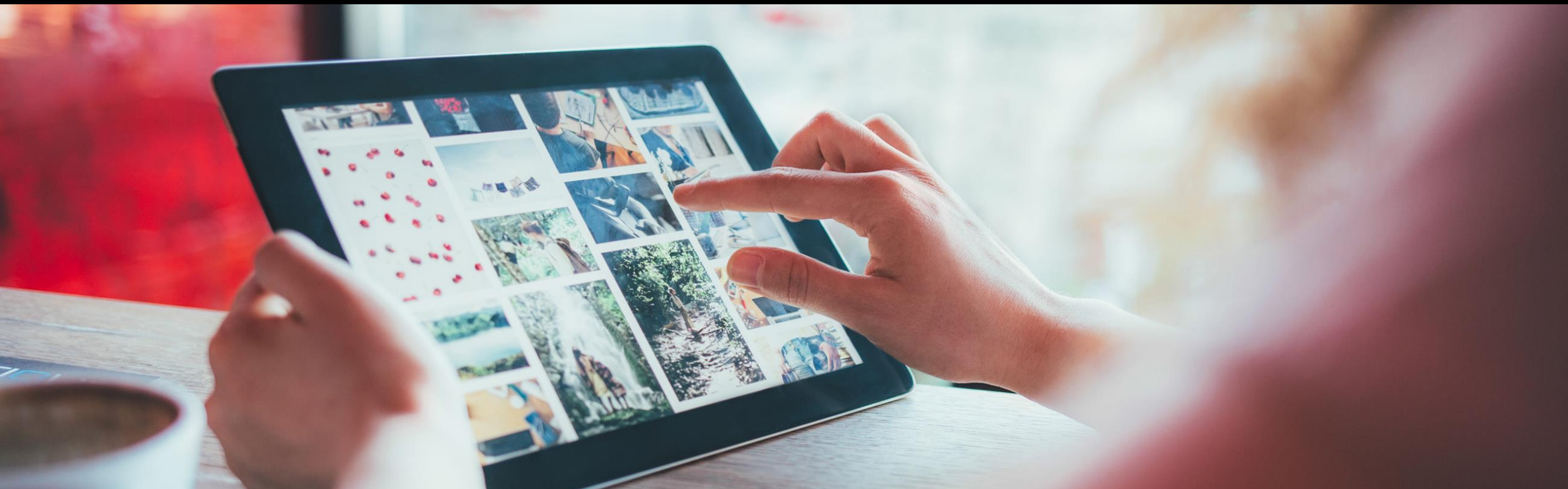
```
public class Main{

    public void pruefeAlter(int alter, int mindestAlter) {
        if(alter >= mindestAlter){
            System.out.println("Volljährig");
        }else{
            System.out.println("Ausweiskontrolle");
        }
    }

    public static void main(String args[]) {
        pruefeAlter(22, 21);
        pruefeAlter(14, 21);
    }

}
```

# Eingabe & Ausgabe



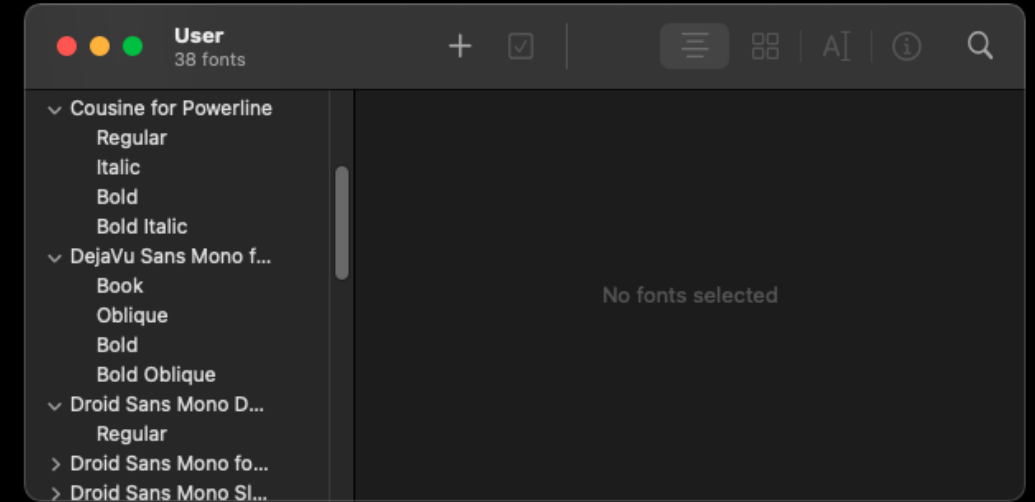
# Eingabe & Ausgabe

## Graphical User Interface (GUI)

- Ein-/Ausgabe mit einer graphischen Oberfläche
- Erleichtert die Bedienung
- Erlaubt die Bedienung auch ohne Kenntnisse über das Programm

### Vorteile

- Viele verschiedene Möglichkeiten der Programmierung (Swing, JavaFX, ...)
- Individuelle Gestaltung
- Erweiterte Ein & Ausgabe (Mausinteraktion, Bilder, Ton, ...)



### Nachteile

- Aufwändig in der Entwicklung
- Fehleranfällig
- Unterschiedlich je nach Betriebssystem

# Eingabe & Ausgabe

## Konsole

- Ein-/Ausgabe über die Konsole
- Erlaubt die Bedienung oftmals nur mit Kenntnissen über das Programm
- Praktisch für Prototypen eines größeren Programms

```
> Player: Michael van Gerwen, 501 points remaining
> Enter visit: 10
> Scored: 10
=====
> Player: Phil Taylor, 501 points remaining
> Enter visit: 20
> Scored: 20
=====
> Player: Michael van Gerwen, 491 points remaining
> Enter visit: _
```

## Vorteile

- Einfach zu Entwickeln
- UI nicht zwingend erforderlich
  - Ressourcensparend
  - Computerlesbar

## Nachteile

- Darstellung begrenzt auf Schriftzeichen
- Wenig Vielfalt in der Ausgabe
- Kaum Interaktionsmöglichkeiten
- Schwierig in der Bedienung

# Eingabe & Ausgabe

## Konsole - Beispiel

```
import java.util.Scanner;

[...]
```

Scanner scanner = new Scanner(System.in);

System.out.print("Gebe deinen Namen ein: ");  
String name = scanner.next(); ←

System.out.print("Gebe eine Ganzzahl ein: ");  
int ganzzahl = scanner.nextInt(); ←

```
> Gib deinen Namen ein: Max  
> Gib eine Ganzzahl ein: 42  
> _
```

# Übung: Input

Schreibe ein Java-Programm, welches einen Text, eine Ganzzahl und eine Gleitkommazahl einliest und danach wieder ausgibt.

```
> Gib einen Text ein: Hello
> Gib eine Ganzzahl ein: 5
> Gib eine Gleitkommazahl ein: 3.1415

> Dein Text: Hello
> Deine Ganzzahl: 5
> Deine Gleitkommazahl: 3.1415

> _
```

## Scanner-Beispiel von der letzten Folie:

```
import java.util.Scanner;

[...]

Scanner scanner = new Scanner(System.in);

System.out.print("Gebe deinen Namen ein: ");
String name = scanner.next();

System.out.print("Gebe eine Ganzzahl ein: ");
int ganzzahl = scanner.nextInt();
```

**Erinnerung:** Eine Gleitkommazahl kann in Java als **double** dargestellt werden.

# Lösung: Input

```
class Main {  
    public static void main(String args[]) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Gib einen Text ein: ");  
        String text = scanner.next();  
  
        System.out.print("Gib eine Ganzzahl ein: ");  
        int ganzzahl = scanner.nextInt();  
  
        System.out.print("Gib eine Kommazahl ein: ");  
        double kommazahl = scanner.nextDouble();  
  
        System.out.println("Dein Text: " + text);  
        System.out.println("Deine Ganzzahl: " + ganzzahl);  
        System.out.println("Deine Kommazahl: " + kommazahl);  
  
    }  
}
```



# Schlussübung

Entwickle ein eigenes Konsolenbasiertes “Text-basiertes” game.

Starthilfe:

```
import java.util.Scanner;

class Main {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Gib deinen Namen ein: ");
        String name = scanner.next();
        System.out.println("Hallo, "+name);

        while(true){
            System.out.println("Du stehst vor drei Türen, welche möchtest du öffnen? [1, 2, 3]");
            int tuer = scanner.nextInt();

            if (tuer == 1){
                System.out.println("Hinter Tür 1 ist nichts.");
            }
            else if (tuer == 2){
                System.out.println("Du öffnest die Tür, und betrittst den Raum. Die Tür fällt direkt hinter dir zu. Was tust du?");
                String wahl = scanner.next();
                System.out.println("Du möchtest "+wahl+"? Super Idee.");
                break;
            }
            else if (tuer == 3){
                System.out.println("Du bist zu schwach um Tür 3 zu öffnen. Versuche es später noch einmal.");
            }
        }
    }
}
```

# Thank you.

Contact information:

**Martin Bratoi**

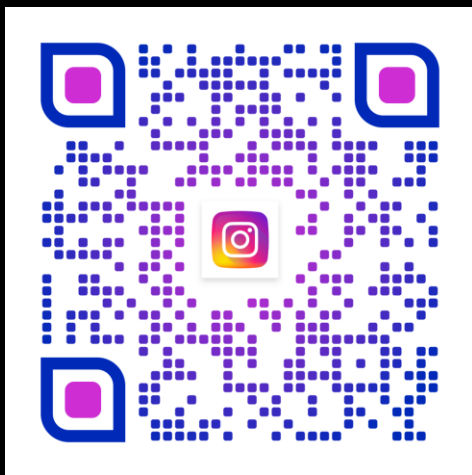
[martin.bratoi@sap.com](mailto:martin.bratoi@sap.com)

**Sebastian Menzel**

[s.menzel@sap.com](mailto:s.menzel@sap.com)



# Follow us on Instagram



[instagram.com/  
sap\\_students](https://www.instagram.com/sap_students)



#VTTakesOff

THE BEST RUN

