



# OOP, DATENTYPEN

GRUNDLAGEN FÜR EIN ERSTES BEISPIEL  
Coders.Eleven - JAVA



## #1 Objektorientiertes Programmieren

Motivation

Klassen/Objekte

Konstruktoren

Getter/Setter

Methoden

## #3 Datentypen

Definition

Arten von Datentypen

## #2 Variablen

Arten von Sprache

Sprache umwandeln

Java Ablauf

## #4 Tipps und Tricks

Ausgabe

Kommentare





# #1 Objektorientiertes Programmieren

Motivation  
Klassen/Objekte  
Konstruktoren  
Getter/Setter  
Methoden



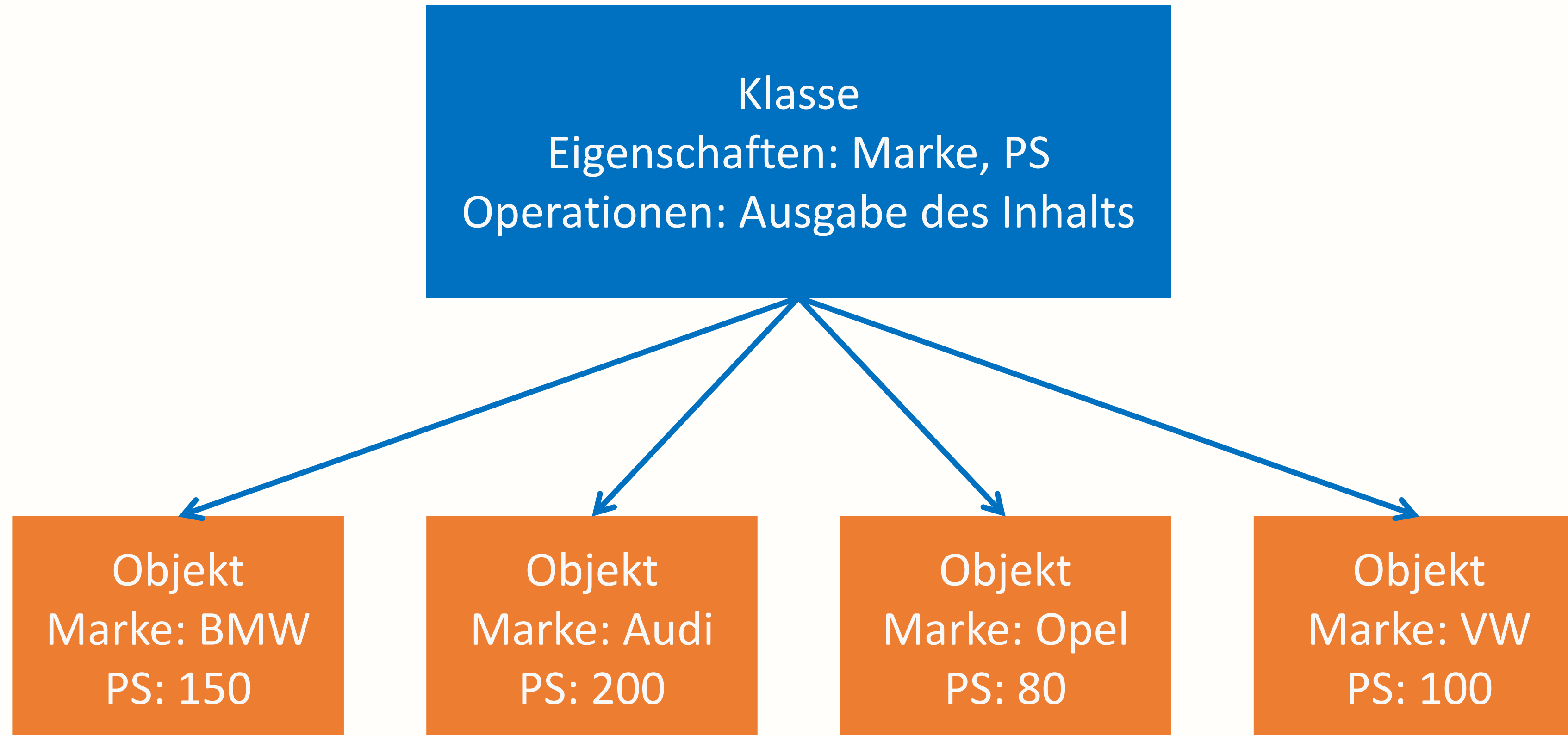
# Motivation

- In der Praxis oftmals große, umfangreiche Programme
  - Viele Funktionen müssen organisiert werden
  - Anforderungen an die Software können sich ändern
- Lösung? Objektorientierung!

# Klassen & Objekte

- Eine Klasse beschreibt die gemeinsamen Eigenschaften und Operationen einer Menge von gleichartigen Objekten
  - Klassen stellen einen „Bauplan“ für Objekte dar
- Ein Objekt ist ein spezifisches Exemplar (Instanz) einer Klasse, welches sich entsprechend dem Bauplan verhält.
  - Objekte sind Datenkapseln (bündeln Daten)
  - Objekte bieten Operationen (Methoden) an

# Beispiel



# Konstruktor

- Der Konstruktor ist eine Initialisierungsmethode, die bei der Erzeugung des Objekts aufgerufen wird
  - Eine Klasse kann mehrere Konstruktoren haben
  - Objektvariablen können damit mit sinnvollen Werten initialisiert werden
- Der Konstruktor
  - hat den gleichen Namen wie die Klasse
  - ist eine Methode ohne Rückgabetyp
  - kann aber Parameter haben

# this-Referenz

- Jedes Objekt hat eine `this`-Referenz
  - Referenz auf sich selbst
- Ist in jeder Methode automatisch vorhanden
- Damit lassen sich Objektvariablen von lokalen Variablen unterscheiden
  - Objektvariablen und lokale Variablen können den gleichen Namen haben



# Default – Konstruktor

- Ein parameterloser Konstruktor wird Default-Konstruktor genannt
  - `public Auto() { ... }`
  - Wird in einer Klasse kein Konstruktor angegeben, fügt Java automatisch einen Default-Konstruktor ein
- Sobald ein Konstruktor angegeben wird, wird kein Default-Konstruktor automatisch erzeugt
  - Wird trotzdem ein Default-Konstruktor benötigt, muss dieser explizit ausprogrammiert werden

# Getter/Setter

- Setter: Methode zur Veränderung einer Objektvariable
- Beispiel:

```
public void setMarke(String marke) {  
    this.marke = marke;  
}
```

Werden in weiterer Folge im  
Konstruktor aufgerufen

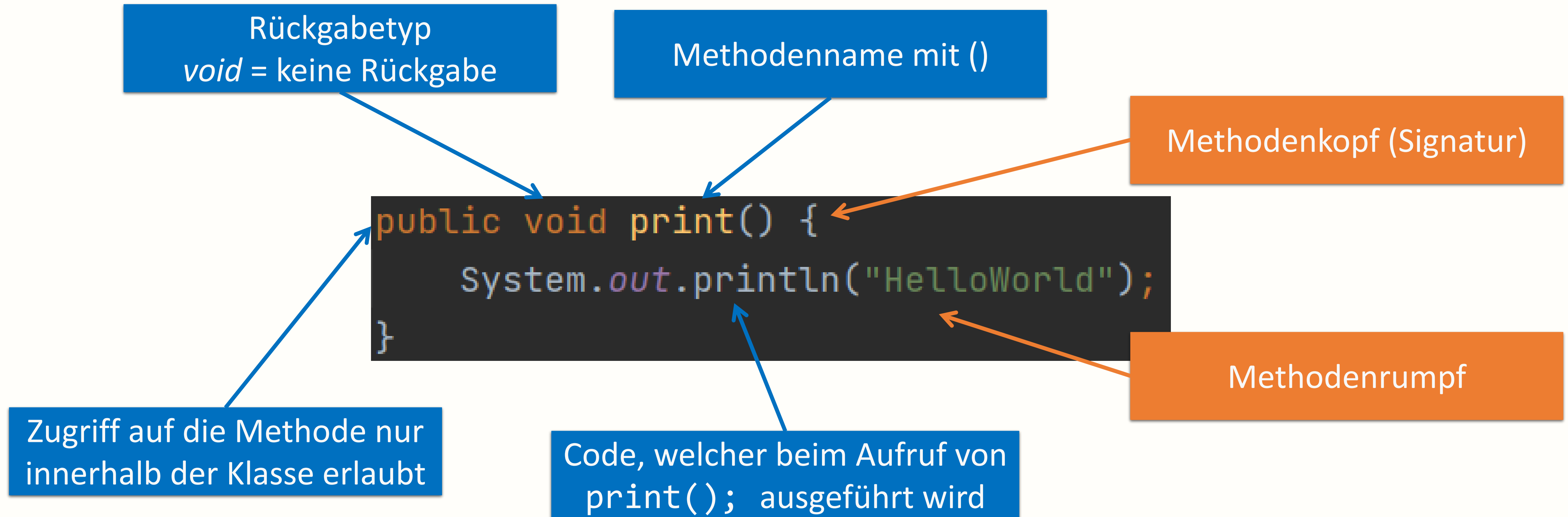
- Getter: Methode, um eine Objektvariable abzurufen
- Beispiel:

```
public String getMarke() {  
    return marke;  
}
```

# Methoden

- Objektbezogene Methoden
  - Gehören zu einem Objekt der Klasse
  - Schauen wir uns momentan an
- Statische Methoden
  - Gehören zur Klasse
  - Schauen wir uns später an

# Aufbau von Methoden



# Rückgabewerte

- Typ des Rückgabewerts steht vor dem Methodennamen
- Wenn „Nichts“ zurückgeliefert wird (zB Methode macht nur eine Ausgabe) ist der Rückgabety `void`
- Ergebnis wird durch `return` zurückgegeben



# #2 Variablen

Deklarieren  
Initialisieren  
Namenskonventionen



# Wofür braucht man Variablen?

- Ein Programm verarbeitet Daten, die in Variablen abgelegt werden
- Ein Programm legt die Ergebnisse wieder in solchen Variablen ab
- Eine Variable ist also eine Speicherstelle
- Kennzeichen einer Variable:
  - Name
  - Datentyp
  - Wert

# Variable deklarieren

- Variable wird einmal vereinbart (deklariert)
  - Name
  - Datentyp
- Beispiel:
  - Variable für eine ganze Zahl mit dem Namen number wird deklariert:

```
int number;
```

Datentyp  
(int steht für ganze Zahlen)

Name  
(für den Zugriff)



# Variable deklarieren & initialisieren

- Variable für eine ganze Zahl mit dem Namen number wird deklariert und mit dem Wert 5 initialisiert:

```
int number = 5;
```

- Variable für eine ganze Zahl mit dem Namen number2 wird deklariert und mit dem Wert der Variable number initialisiert:

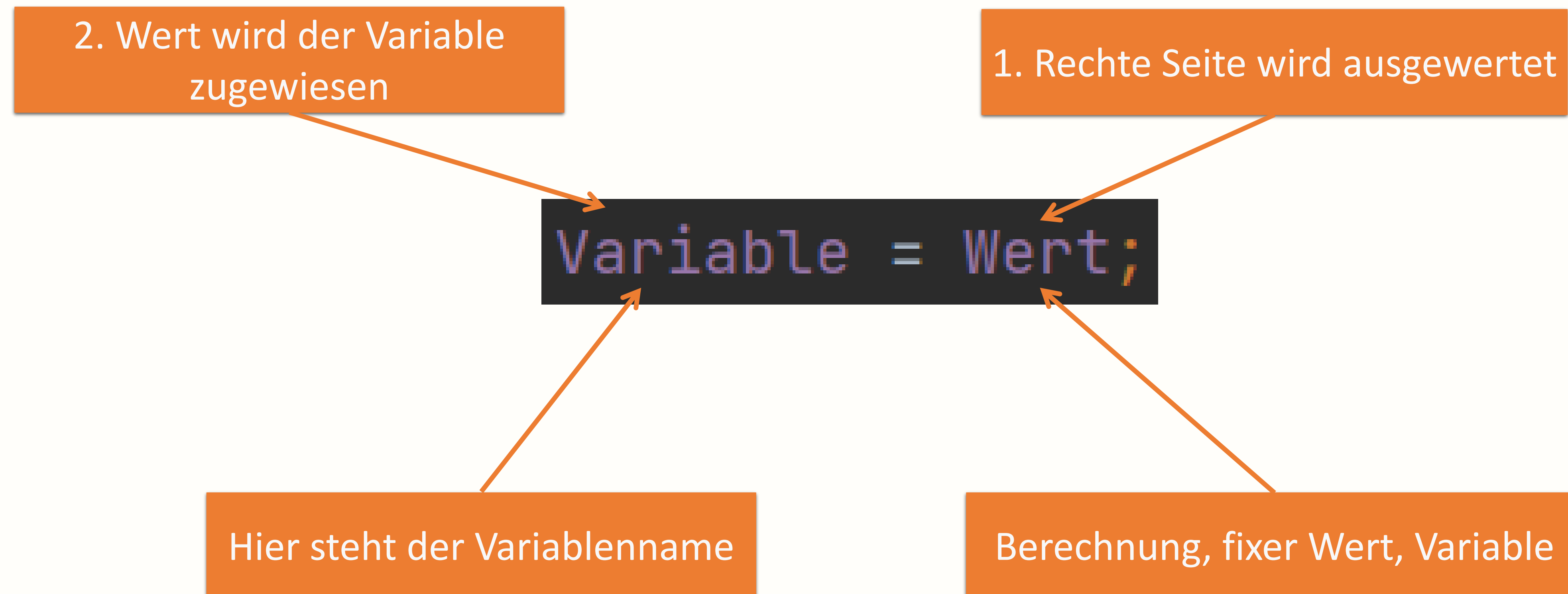
```
int number2 = number;
```

- Variable für eine ganze Zahl mit dem Namen number3 wird deklariert und mit dem Wert 25 initialisiert:

```
int number3 = 10 + number * 3;
```

# Wert zuweisen

- Angenommen die Variable wurde bereits deklariert
- Dann kann separat ein Wert zugewiesen werden:



# Namenskonventionen

- Namen sollten gewissen (Java-) Konventionen folgen:
  - Kleiner Anfangsbuchstabe
  - Lower Camel-Case
  - Keine Zahl als erstes Zeichen
  - Keine Sonderzeichen außer \$ und \_
  - Aussagekräftige Namen (außer bei Hilfsvariablen)
  - Keine Keywords (Schlüsselwörter)
  - Vorzugsweise Englische Bezeichnungen

# #3 Datentypen

Definition

Arten von Datentypen



# Was ist ein Datentyp?

- Daten haben einen bestimmten Typ (Datentyp)
- Datentyp definiert:
  - Wertebereich
  - Operationen
- Compiler führt vor Ausführung des Programms eine Überprüfung durch

# Arten von Datentypen

#3 Datentypen

Primitive Datentypen				
Name	Speicherplatz	Typ	Wertebereich	Defaultwert
boolean	1 Bit	Wahrheitswert	true/false	false
byte	1 Byte	ganze Zahlen	-128 bis 127	0
short	2 Bytes	ganze Zahlen	-32.768 bis 32.767	0
char	2 Bytes	ein Zeichen		'\u0000'
int	4 Bytes	ganze Zahlen	-2.147.483.648 bis 2.147.483.647	0
float	4 Bytes	Gleitkommazahl	-3,4028234663852886e+38 bis 3,4028234663852886e+38	0.0f
long	8 Bytes	ganze Zahlen	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807	0L
double	8 Bytes	Gleitkommazahl	-1,7976931348623157e+308 bis 1,7976931348623157e+308	0.0d
Nicht primitive Datentypen				
String	Variabel	Zeichenkette		null

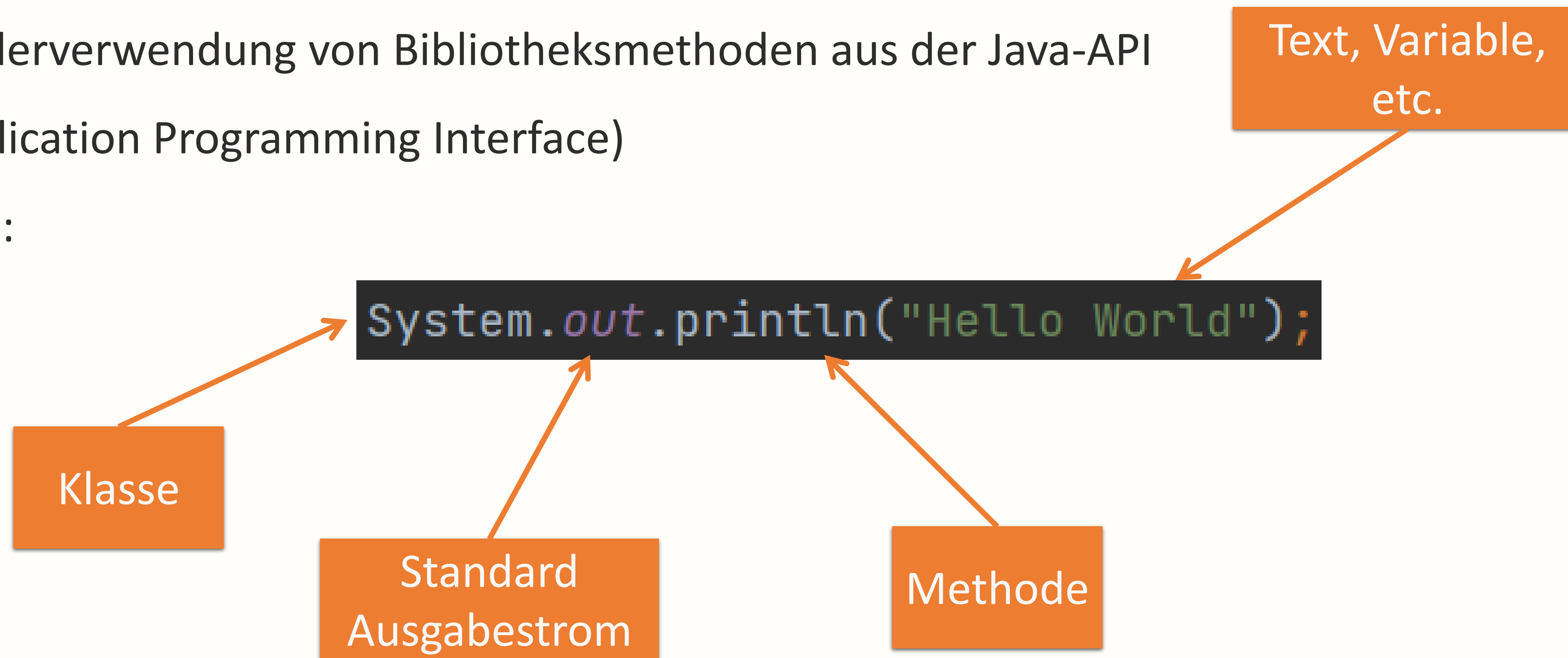
# #4 Tipps und Tricks

Ausgabe  
Kommentare



# Ausgabe in der Konsole (1)

- Ausgabe muss nicht selbst programmiert werden
  - Wiederverwendung von Bibliotheksmethoden aus der Java-API (Application Programming Interface)
- Aufbau:





# Ausgabe in der Konsole (2)

- Beispiele für Varianten

```
System.out.print(4);  
System.out.print("Hello");  
System.out.print(a);  
System.out.println(a);  
System.out.println();  
System.out.print(4 + 2);  
System.out.print("Hello" + 2);
```

Verknüpfung von  
Ausdrücken mit  
dem + Operator

4Hello55

6Hello2

# Kommentare in Java

- Dienen zum Beschreiben des Codes, werden nicht übersetzt

- Formen:

- Einzeilig

```
// Einzeiliger Kommentar
```

- Mehrzeilig

```
/*  
Mehrzeiliger Kommentar  
*/
```

- Dokumentation

```
/**  
 * Java Dokumentation  
 */
```