

# Programmiervorkurs

## Wintersemester 20/21

Tag 1



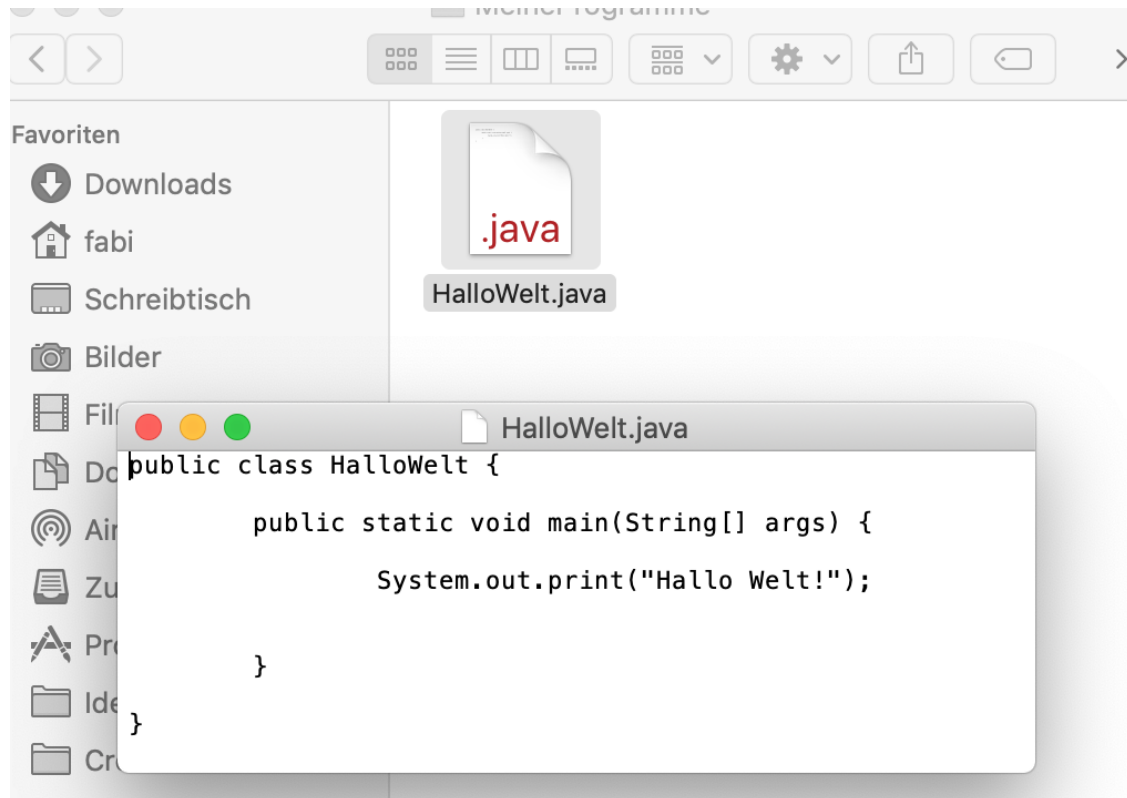
# Agenda

- › Einführung
- › Installation und erste Schritte mit den Werkzeugen
- › Tutorium
  - › Ausgaben auf der Konsole
  - › Umgang mit Variablen
  - › Objektorientierung (Teil 1)

# Was ist Programmieren?



## › Programmiersprache

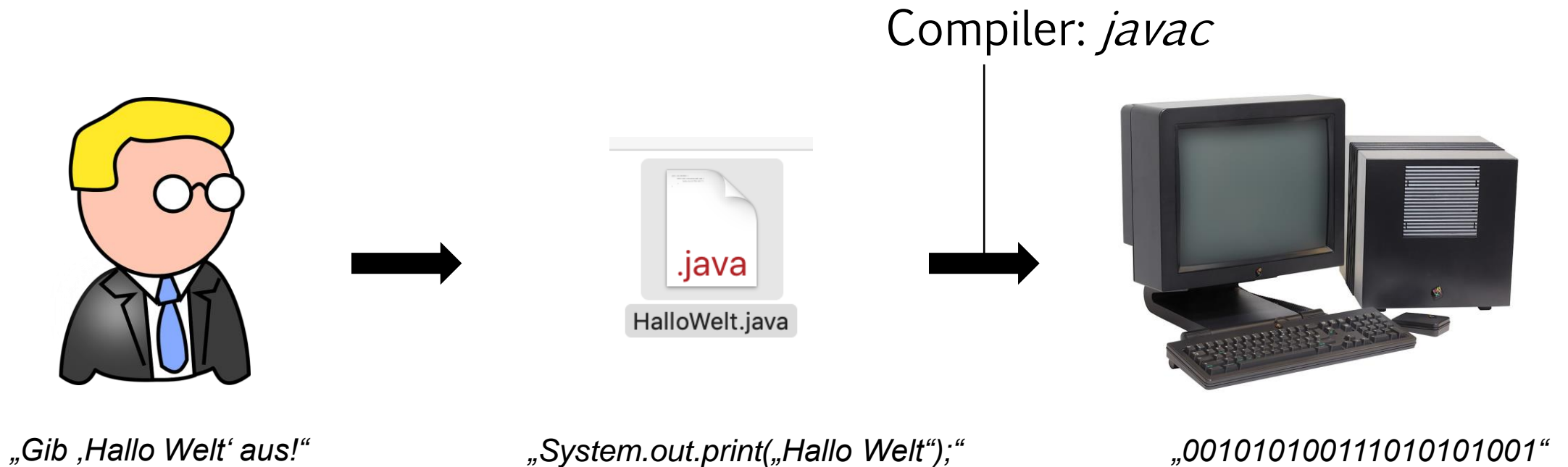


- › Existiert auch *ohne* irgendwelche Tools (Coden im Texteditor meiner Wahl)
- › Ähnlich wie menschliche Sprache: „Vokabeln“ + „Grammatik“ + Übung macht den Meister
- › Wie kann der Computer diese Sprache verstehen?

# Was ist Programmieren?



- › Wie kann der Computer diese Sprache verstehen?



- › Woher bekomme ich den Compiler?

Anmerkung: Dieser Prozess ist sehr vereinfacht dargestellt

# Was ist Programmieren?



- › JDK = Java Software Development Kit

- › Download:

- <https://www.oracle.com/technetwork/java/javase/downloads/jdk12-downloads-5295953.html>

- › Enthält den Compiler

- › Enthält Dokumentation (ausführliche „Anleitung“)

- › Enthält viele nützliche zusätzliche Pakete fürs Programmieren

# Was ist Programmieren?



1. Quelltext schreiben (in Java)
2. Quelltext kompilieren (mit javac)
3. Ggf. Kompilierfehler im Quelltext beheben (zurück zu 2.)
4. Programm ausführen

# Was ist Programmieren?



1. Quelltext schreiben (in Java)
2. Quelltext kompilieren (mit javac)
3. Ggf. Kompilierfehler im Quelltext beheben (zurück zu 2.)
4. Programm ausführen

1. Quelltext schreiben (in Java)
2. Programm ausführen



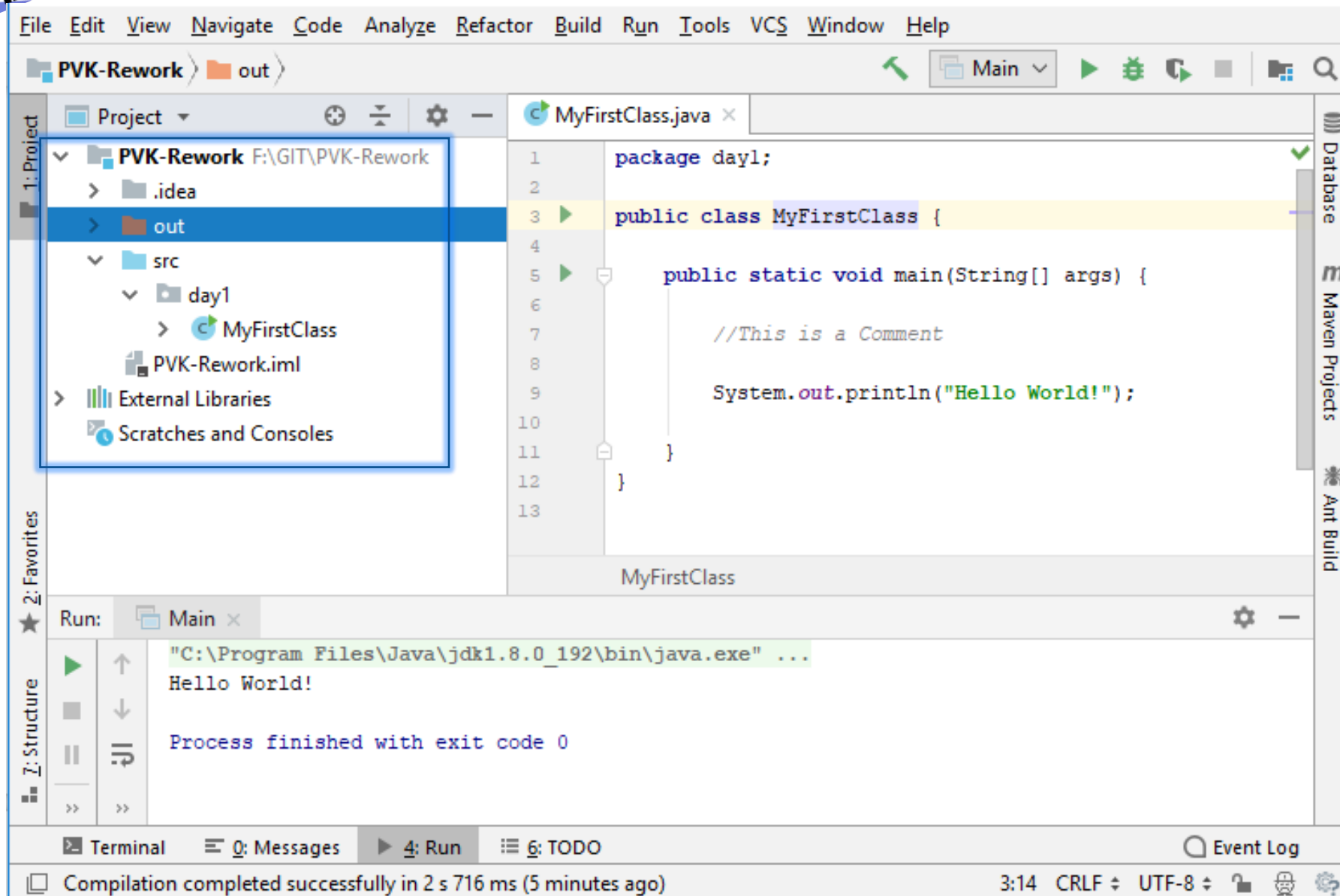
# Was ist Programmieren?



- › IDE = Integrated Development Environment
- › Download: <https://www.jetbrains.com/idea/download/>
- › Übernimmt den Kompilierungsprozess
- › Zeigt Kompilierfehler direkt beim Schreiben des Quelltextes an
- › Schöne Farben („Syntax Highlighting“)
- › Nützliche Hinweise und Warnungen zum Code
- › Eingebauter Debugger (für Logikfehler im Programm)
- › ...



# Was ist Programmieren?



# Ausgaben auf der Konsole

- › Mit Zeilenumbruch:

```
System.out.println("Ausgabertext");
```

- › Ohne Zeilenumbruch

```
System.out.print("Ausgabertext");
```

- › Manuelle Textstrukturierung

- › Neue Zeile `\n`

- › Tabulator `\t`

# Aufgabe: Ausgabe von Text

## GRUNDLAGE

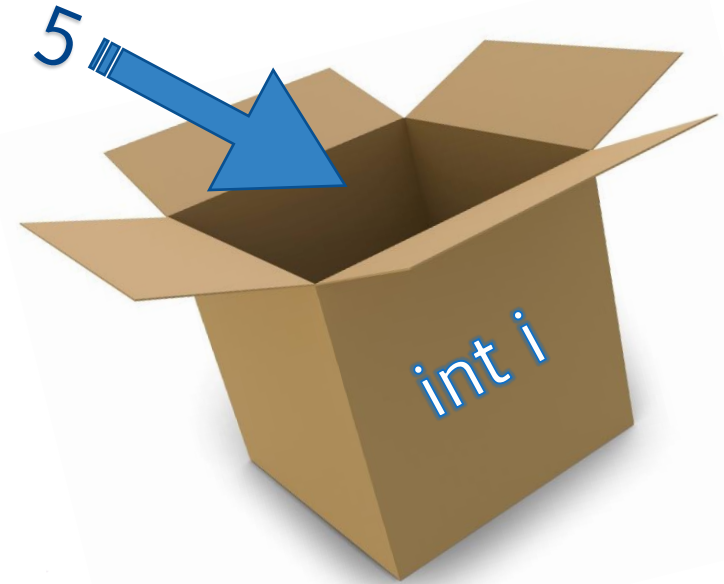
- › Schreibt ein Programm mit neuem Projekt, Package und Klasse, das folgende Sätze auf der Konsole ausgibt:
  - › Hello World!
  - › Hello World!  
How are you?

## ZUSATZ

- › Schreibe ein Programm, dass den Backslash (\) auf der Konsole ausgibt.

# Variablen

- › Typen
  - › Zeichenkette
    - › **String**
  - › Ganze Zahlen (z.B. 2)
    - › byte
    - › short
    - › **int**
    - › long
  - › Fließkommazahlen (z.B. 2,5)
    - › float
    - › **double**
  - › Weitere Datentypen
    - › **boolean** (Wahrheitswert, true/false)
    - › char (ein Zeichen, z.B. a, b, ...)



# Variablen



Typ	Minimaler Wert	Maximaler Wert	Anzahl möglicher Werte
byte	-128	127	$2^8$
short	-32768	32767	$2^{16}$
int	-2147483648	2147483647	$2^{32}$
long	-9223372036854775808	9223372036854775807	$2^{64}$
float	1.4012984643248171E-45	3.4028234663852886E+38	$2^{32}$
double	4.9406564584124654E-324	1.7976931348623157E+308	$2^{64}$

# Variablen

- › Deklaration:

Datentyp	Bezeichner	;
<i>int</i>	<i>i</i>	;

- › Zuweisung:

Bezeichner	=	Wert	;
<i>i</i>	=	<i>2</i>	;

- › Ziel: Festlegen des Datentyps und spätere Zuweisung
- › Variablen können später mit neuen Werten überschrieben werden.

# Variablen

## › Deklaration und Zuweisung

Datentyp	Bezeichner	=	Wert	;
<i>int</i>	<i>i</i>	=	2	;
<i>double</i>	<i>geldbetrag</i>	=	2.5	;
<i>boolean</i>	<i>b</i>	=	<i>true</i>	;
<i>String</i>	<i>name</i>	=	"Tina"	;

## › Bezeichner beliebig wählbar (mit Einschränkungen)

- Guter Stil: beginnt mit Kleinbuchstaben und ist selbsterklärend
- Erlaubt: `hallo`, `_hallo123`, `Hallo_Welt`, `H_A_L_L_O`
- Nicht erlaubt: `2und2macht4`, `Hallo Welt`, `class`, `null`, `Faster!`

# Variablen

- › Ausgabe wie bei normalem Text

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

- › Auswertung von rechts nach links

```
int anzahlStudenten = 3 + 4; → Ausgabe: 7
```

```
anzahlStudenten = anzahlStudenten + 4; → Ausgabe: 11
```

- › Mathematische Operatoren

+      -      \*      /



# Variablen

## › Ausgabe von Text und Variablen (Konkatenation)

```
System.out.println("Anzahl der Studenten:" + anzahlStudenten);
```

### Addition

```
int i = 3 + 4;
```

### Konkatenation

```
System.out.println("Ergebnis: " + i);
```

→ Ausgabe: Ergebnis: 7

# Aufgabe: Arbeiten mit Variablen

## GRUNDLAGE

- › Erstelle ein `int i` mit Wert 4.
- › Addiere 3 dazu.
- › Erstelle ein `String s` mit Wert `"7"`
- › Nutze die zwei Variablen um auf der Konsole `"7 = 7"` auszugeben.

## ZUSATZ

- › Erstelle ein `double` mit Wert 1,0
- › Erstelle ein `double` mit Wert 2,5
- › Erstelle ein `String` mit Wert `"1,0 + 2,5 = "`
- › Nutze die 3 Variablen um `"1,0 + 2,5 = 3.5"` auf der Konsole auszugeben OHNE außerhalb von `System.out` zu rechnen und OHNE 3.5 einzugeben.

# Objekt

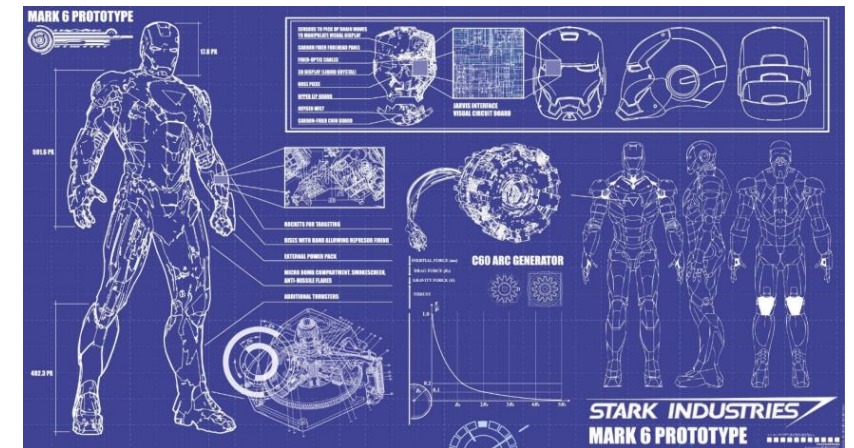
- › “Container“, der alle Eigenschaften (=Attribute) von Gegenständen, Personen, Visitenkarten etc. umfasst
- › Beispiele:



- › Attribute haben “Variablennamen“ und Datentypen  
→ vom Programmierer festgelegt

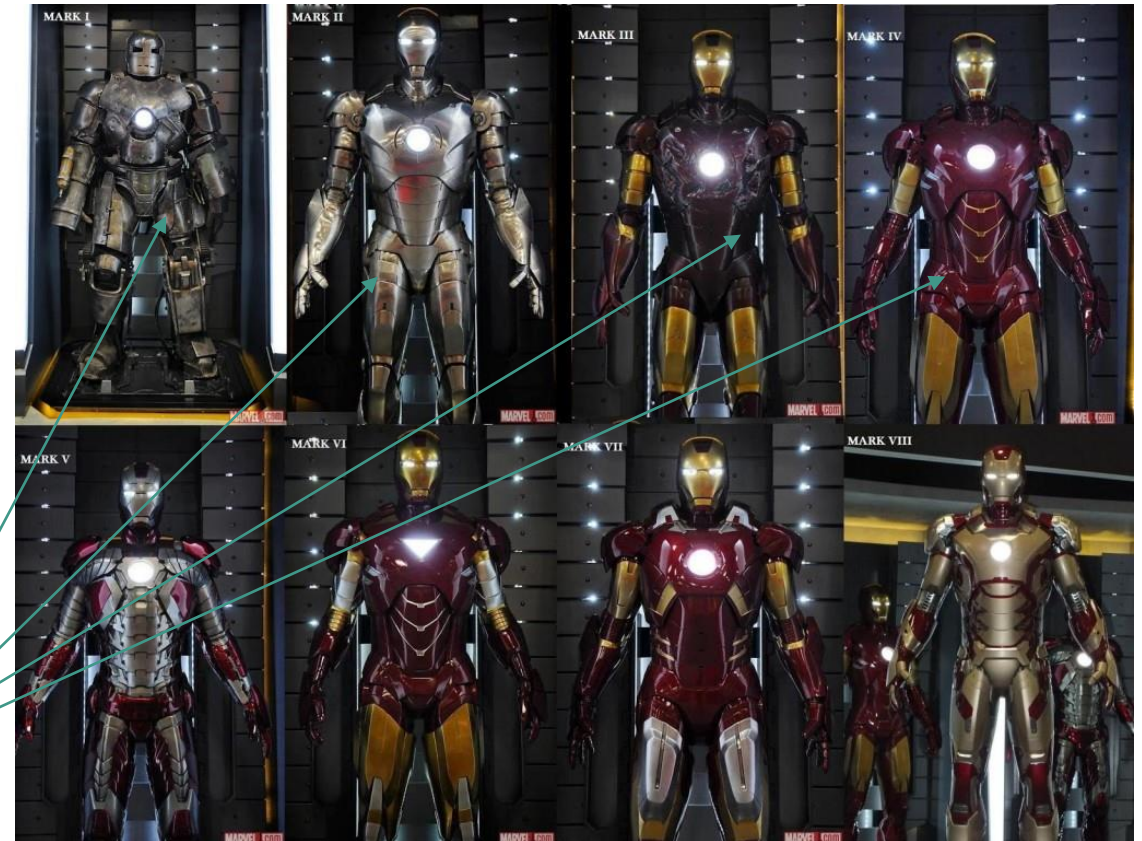
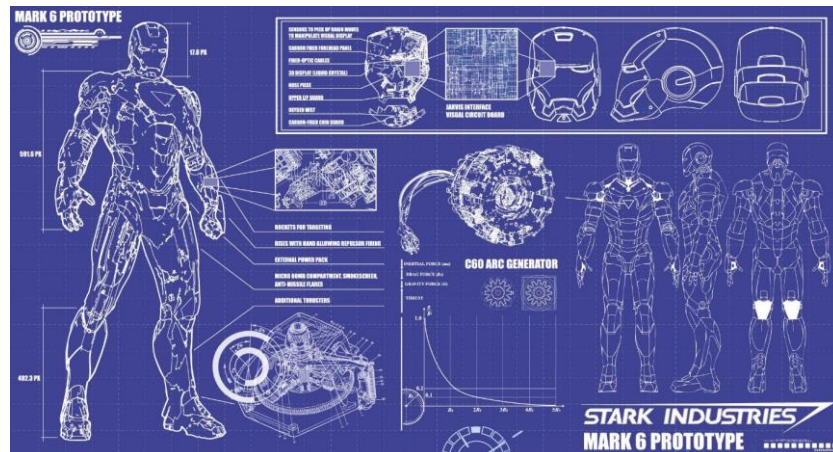
# Klassen

- › Schablone, die man beliebig oft verwenden kann
- › Legt mögliche Attribute fest, die man belegen kann
- › Feste Definition, dass Rechteck ein Attribut ist und es niemand `seiteA` oder `kante1` nennt



# Erstellen von Objekten

- › Von einer Klasse kann ich beliebig viele Objekte erstellen, die sich aber in ihren Eigenschaften unterscheiden können





# “Das Objekt der Klasse ...”

- › *„Fiffi ist ein Objekt der Klasse Hund“*
  - › Name: Fiffi
  - › Rasse: Chihuahua
  - › Fellfarbe: Schwarz
  - › Schulterhöhe in cm: 11
- › *“Mein BMW ist ein Objekt der Klasse Auto“*
  - › ...

# Erstellen von Objekten

- › Objekte werden ähnlich wie Variablen erzeugt

```
Rechteck  
Rechteck  
r1 = new  
r2 = new  
Rechteck()  
Rechteck();
```

Klassenname

(Schablone, die verwendet wird)

New-Operator: neues Objekt erzeugen

Variablennamen mit dem das Objekt  
angesteuert werden kann

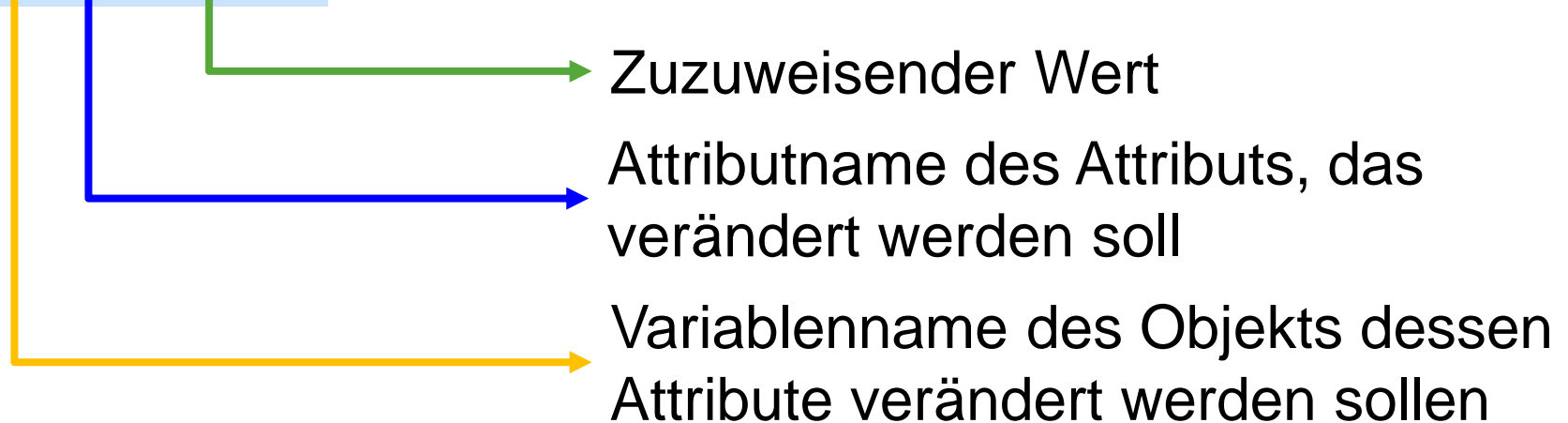
Typ des neuen Objekts  
(Schablone, die verwendet wird)

# Zugriff und Verwendung von Objekte

- › Zugriff auf Attribute eines Objekts

```
r1.a = 3;  
r1.b = 4;
```

Zuweisung



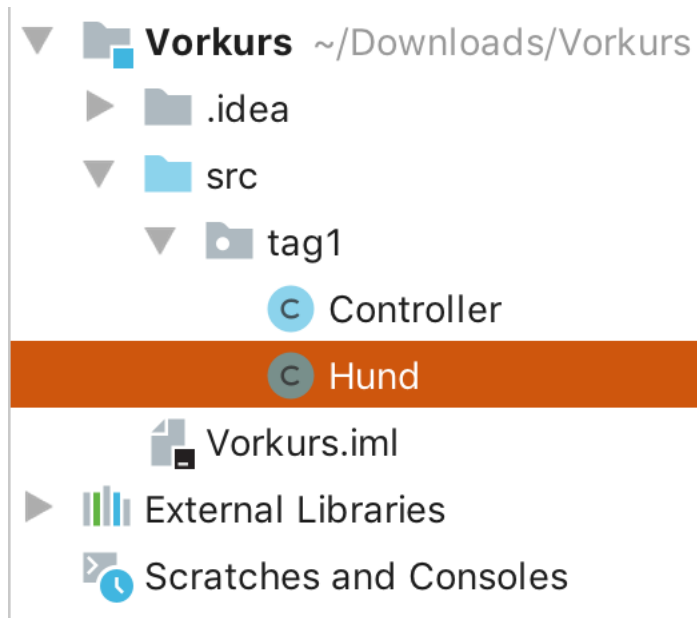
- › Ausgabe eines Attribut-Wertes

```
System.out.println("Seitenlänge von Rechteck a: " + r1.a);
```



# Dateiverwaltung von Klassen

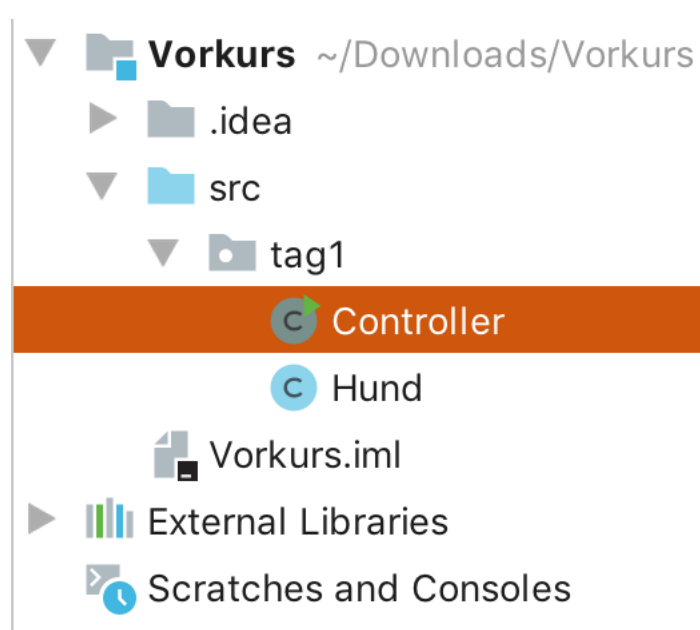
- › Wie erstelle ich eine Klasse in IntelliJ?



```
public class Hund {  
  
    String name;  
    String rasse;  
    String fellfarbe;  
    double hoehe;  
  
}
```

# Dateiverwaltung von Klassen

- › Wie erstelle ich ein Objekt aus einer Klasse in IntelliJ?



```
public class Controller {  
  
    public static void main(String[] args) {  
  
        Hund h1 = new Hund();  
        Hund h2 = new Hund();  
  
        h1.name = "Fiffi";  
        h2.hoehe = 12.1; //cm  
  
    }  
}
```

# Übungen aus der Aufgabensammlung

Aufgabe	Seite	Schwierigkeit (1-10)	Freiheitsgrad (0-3)
Ausgabe von Text	5	1	0
Ascii Katze	6	2	0
Arbeiten mit Variablen	7	1	0
Visitenkarte	8	2	0

# Abschlussaufgabe: Ausgabe von Variablen

- › Schreibt ein Programm, das Eure Visitenkarte ausgibt. Legt dazu Eure Daten in einzelnen Variablen an.
- › Tipp: Überlegt Euch vorher, wie viele Variablen Ihr braucht, welchen Datentyp sie haben und wie Ihr sie benennt!
- › Ausgabe-Beispiel:
  - › Name: Erika Mustermann
  - › Straße: Musterstraße 13
  - › Ort: 12345 Musterstadt
  - › Telefonnummer: 0123 456789
  - › Geburtsdatum: 01.01.1970
  - › Alter: 47