

# Einführung in die Programmierung für Nebenfach Medieninformatik

Beat Rossmay, Michael Kirsch



- Eure Mitarbeit ist uns wichtig!
- Installiert Euch dazu die kostenlose App „Socrative Student“ auf Eurem Smartphone oder nutzt das Webinterface unter <http://b.socrative.com/login/student/>
- Anonymer Login über den Raumnamen:

***MSMJOKRQ***





## Frage 1

Wie zufrieden seid Ihr mit dieser Vorlesung und den Übungen?

A:

B:

C:

D:



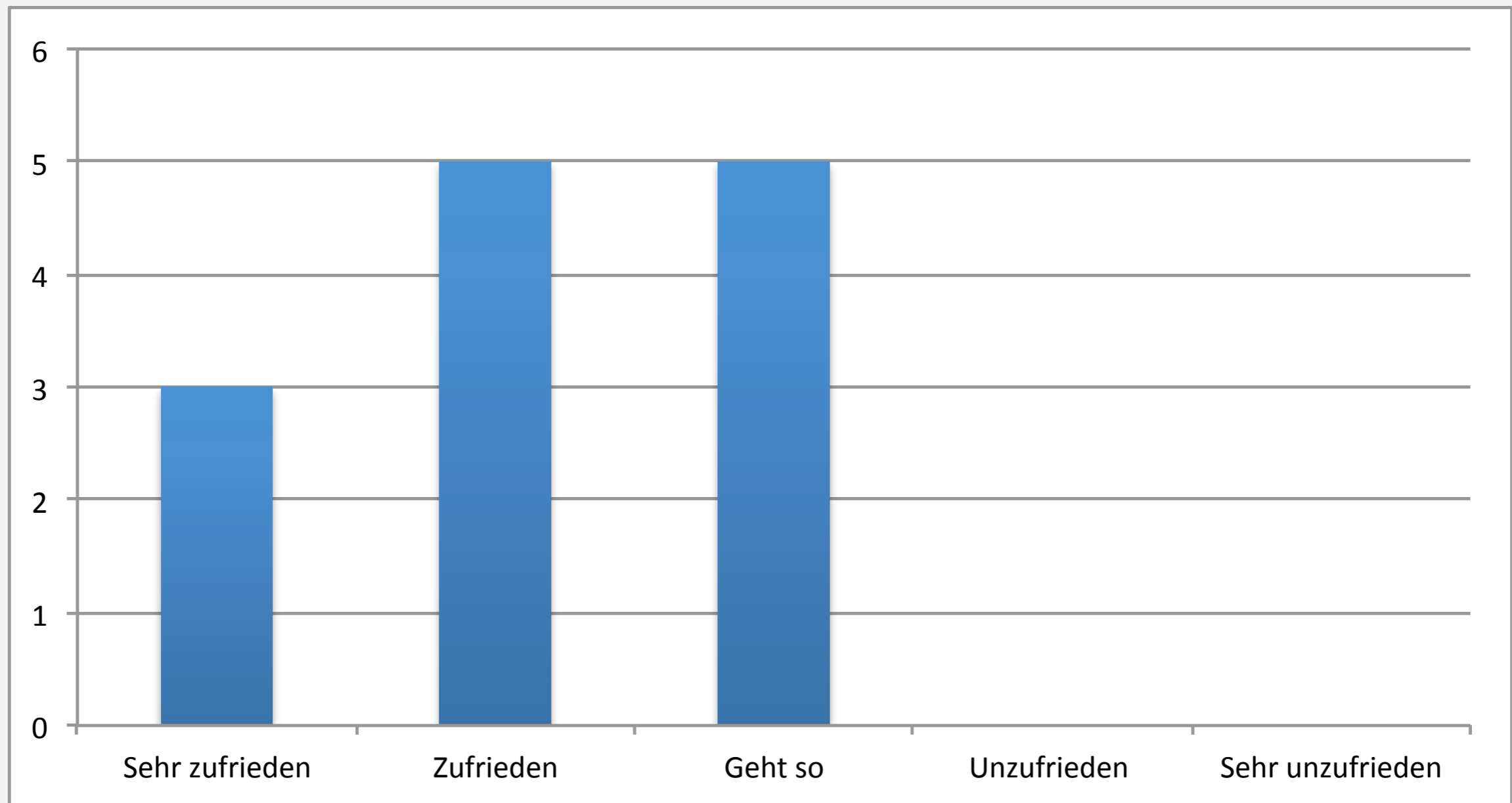
*MSMJOKRQ*

Klausurtermin am 23.02.2018

Uhrzeit: 12-14 Uhr (s.t.)

Raum: Hauptgebäude LMU M 018

Frage 1: Wie zufrieden seid Ihr mit dieser Vorlesung und den Übungen? (Ergebnisse aus dem letzten Semester)



# Einführung in Java

Beat Rossmay, Michael Kirsch



## I. Einleitung

1. Wie geht es weiter?
2. Java & Processing
3. Java Beispiele

## II. Theorie

1. IntelliJ
2. Aufbau von Java
3. Schlüsselwörter
4. Kontrollstrukturen
5. Hello World!

## III. Anwendung

1. Java in Action

## IV. Verknüpfung

1. Processing in Java
2. Videotutorials

## V. Ausblick

1. Nächste Vorlesung
2. Übung

# Einleitung



## Einführung in die Programmierung

### Processing

Typen und  
Operatoren

Kontroll-  
Strukturen

Klassen und  
Objekte

Gültigkeit und  
Konventionen

Methoden

Arrays

Konstruktoren

Eingaben-  
verarbeitung

Animationen

...

### Java

Grundlagen aus  
Processing

...



Processing

Java



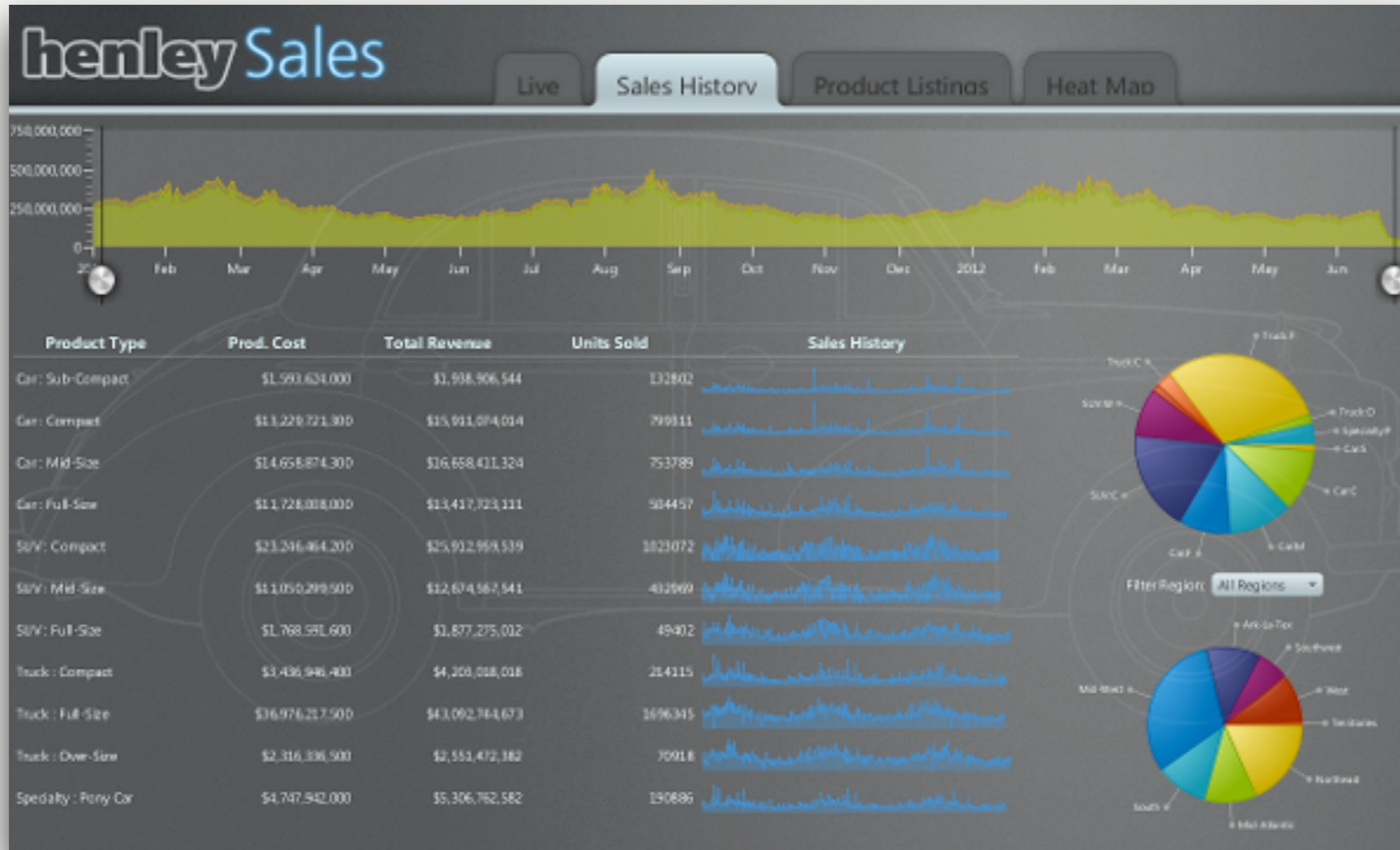
- Java wurde 1995 von Sun Microsystems entwickelt und 2010 von Oracle gekauft und seitdem unter dem Namen „Oracle Java“ geführt
- Java ist eine Programmiersprache UND eine Technologie
- Java war bei der Entwicklung von Processing das Vorbild
  - Jedoch sollte Processing leichter zu verwenden sein
  - Zielgruppe von Processing sind „Visual Designers“
  - Zielgruppe von Java sind Softwareentwickler

- Warum verwenden wir nicht weiterhin Processing?
  - Processing ist mächtig, aber dennoch begrenzt
  - Komplexere Anwendung bedürfen einer komplexeren Sprache und einer mächtigen Entwicklungsumgebung, die uns unterstützt
- Java bietet viele vorgefertigte Komponenten und erleichtert uns die Entwicklung von komplexeren Anwendung (Bspw. ein Spiel)



## Interactive Audio Player

<http://cache.fxexperience.com/wp-content/uploads/2012/01/FXExperiencePlayer-BG.png>

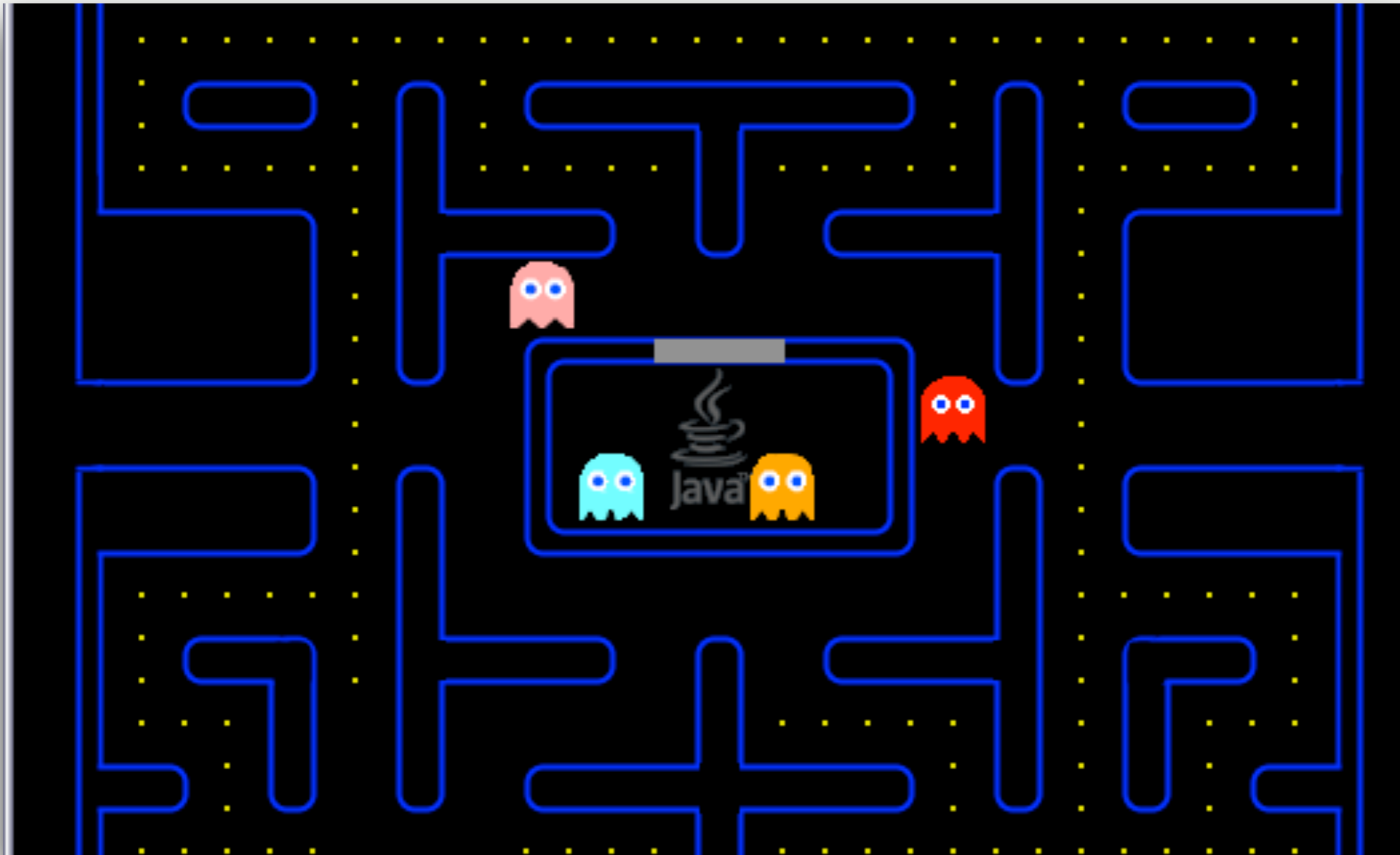


## Fancy Charts

[https://docs.oracle.com/javafx/2/best\\_practices/jfxpub-best\\_practices.html](https://docs.oracle.com/javafx/2/best_practices/jfxpub-best_practices.html)



Fancy Games - Brick Breaker



## Fancy Games - Pacman

<http://www.javafxgame.com/screenshot.png>



## Benutzerdialoge

[docs.oracle.com](https://docs.oracle.com)





The screenshot shows the GORTZ website interface for women's shoes. At the top, there is a navigation bar with the GORTZ logo, service information (free shipping and returns), a customer rating of 'SEHR GUT', and a shopping cart icon showing 0 items for €0.00. Below the navigation bar, there are category tabs: NEU, SCHUHE, TASCHEN, ACCESSOIRES, TRENDNEWS, MARKEN, and SALE %. A search bar is also present.

The main content area is titled 'Damenschuhe (2.239 Artikel)'. It features a filter bar with dropdown menus for 'Kategorie', 'Farbe', 'Marke', 'Grösse', and 'Absatz'. The sorting is set to 'Beliebtheit'. There are pagination controls showing pages 1, 2, 3, and 22.

The product grid displays several items:

- CONVERSE CTAS DAINTY OX**: € 64,95
- COX Riemchen-Sandalette**: € 49,95
- TOMMY HILFIGER Sandalette ELBA 19D**: € 79,95
- VAGABOND Sandalette MINHO**: € 69,95
- SALE TOMMY HILFIGER Sandalette VERONICA**: ~~€ 89,95~~ € 69,95 (-22%)
- HILFIGER DENIM**: (Image of a dark blue sandalette)
- PAUL GREEN**: (Image of a dark blue sneaker)
- AKIRA**: (Image of a brown sandalette)
- RAINBOW CLUB**: (Image of a white high-heeled pump)

## Industrial Webservices

[www.goertz.de/](http://www.goertz.de/)



„Ich bin aber kein  
Softwareentwickler...“

























„Du wirst aber mit welchen  
zusammen arbeiten (müssen)!“

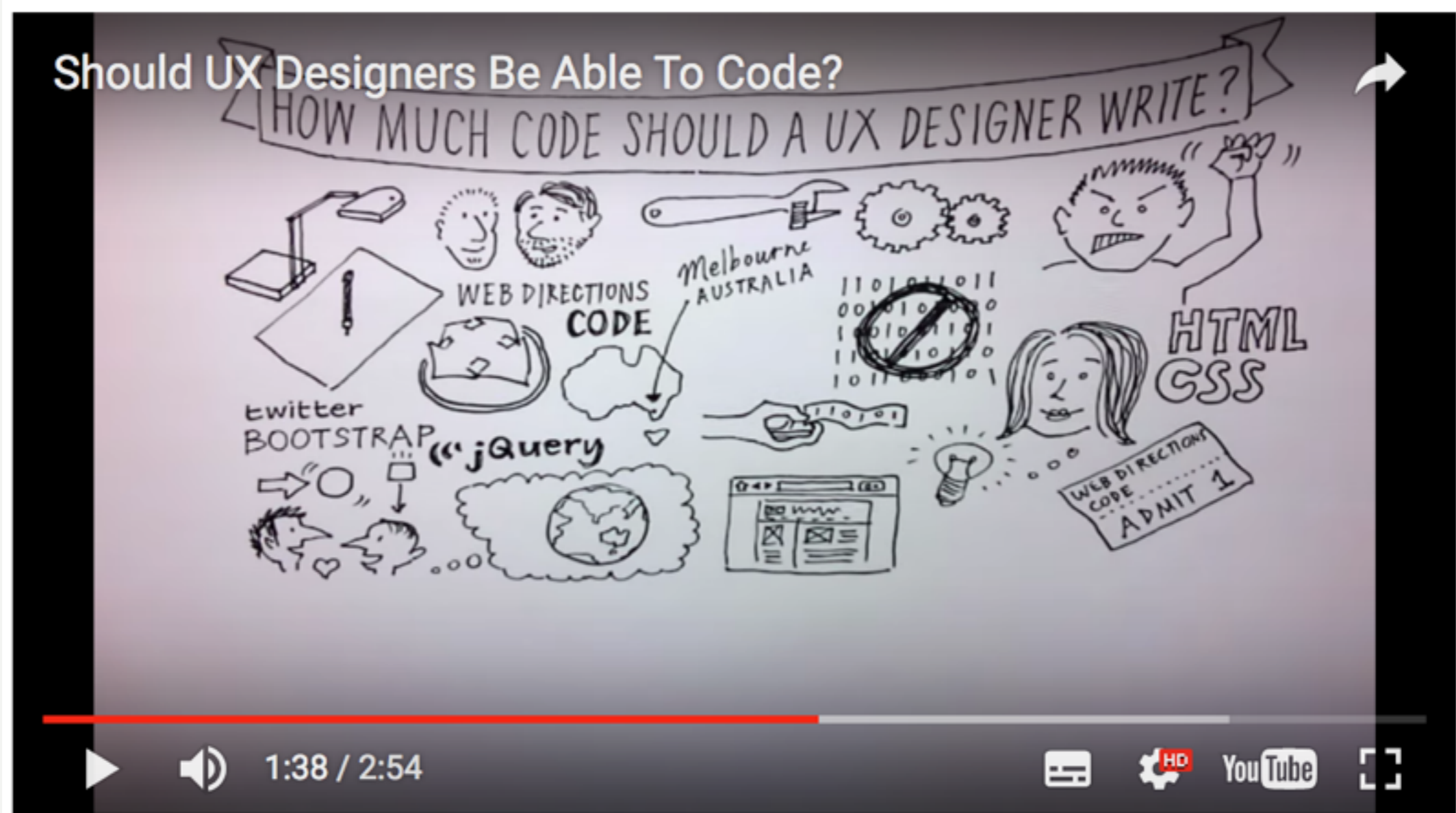


Verbreitung von Programmiersprachen

# The 2017 Top Ten Programming Languages



Language Rank	Types	Jobs Ranking
1. Java	  	100.0
2. C	  	99.4
3. Python	 	99.3
4. C++	  	92.2
5. JavaScript	 	89.9
6. C#	  	86.4
7. PHP		80.5
8. HTML		79.7
9. Ruby	 	76.6
10. Swift	 	76.4



<https://entwickler.de/online/development/7-gruende-coding-skills-designer-297235.html>



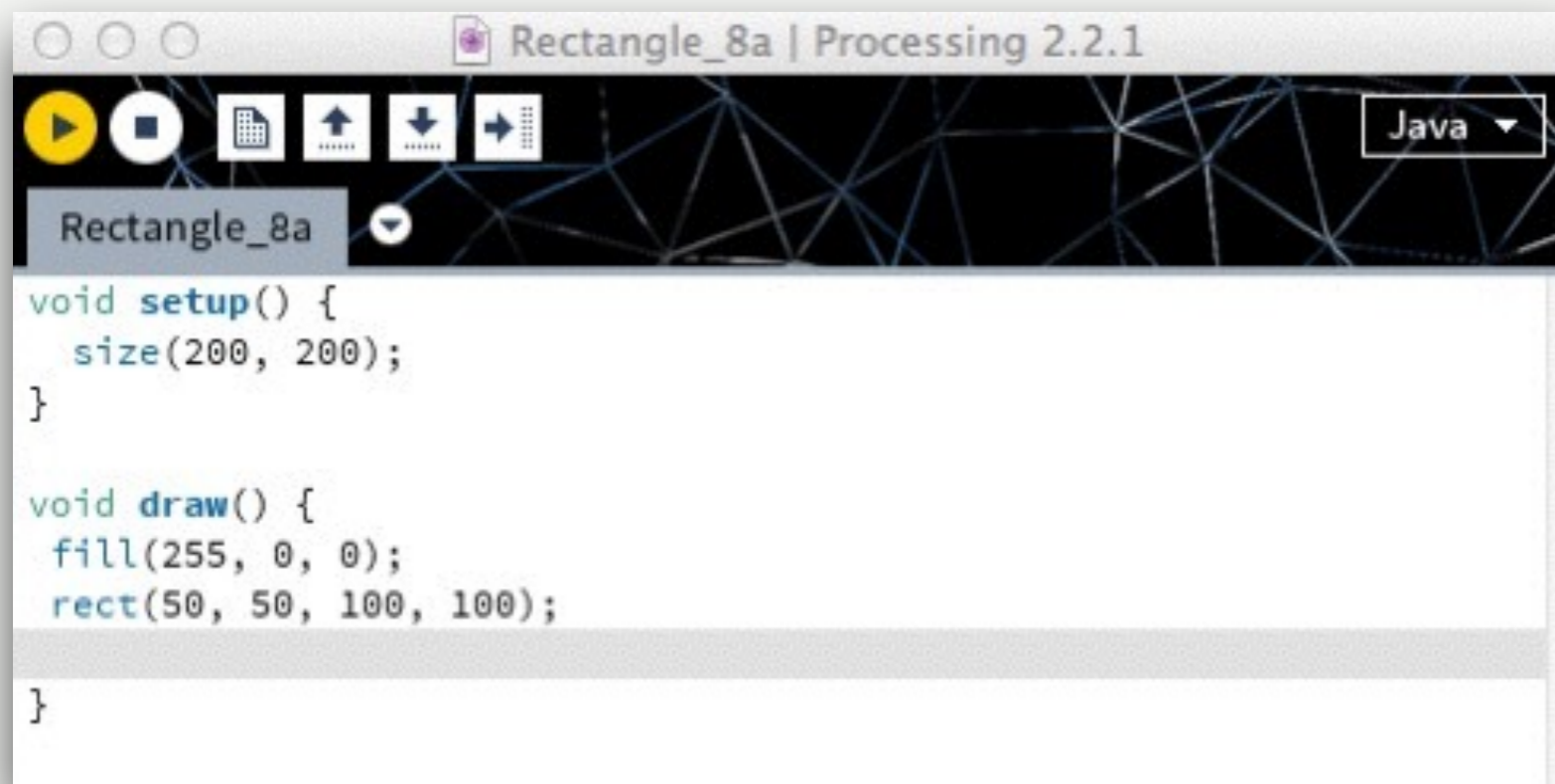
# Theorie



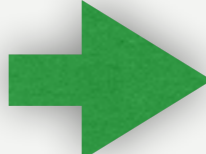
- Wie sieht Java aus?

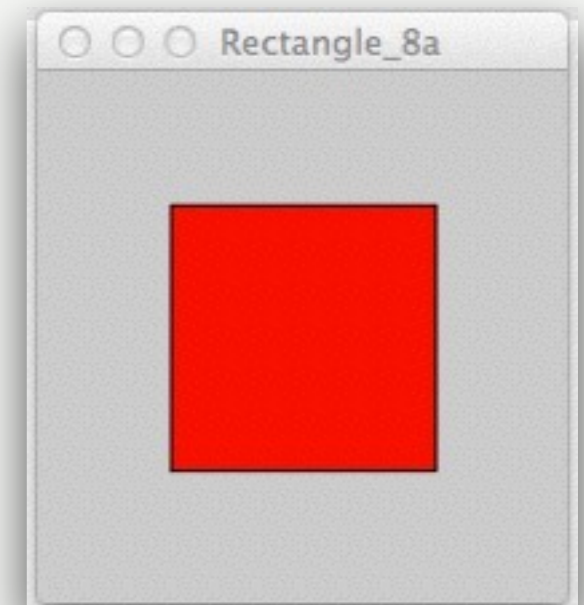
```
public class HelloWorld {  
  
    public static void main(String[] args) {  
  
        for (int i = 0; i < 10; i++) {  
  
            System.out.println(" Java is so cool! ");  
  
        }  
  
    }  
  
}
```

- Da Processing Java als Vorbild dient, ist es relativ leicht möglich Euer bisheriges Wissen von Processing auf Java zu übertragen...
- In Processing wird ein rot ausgefülltes Quadrat wie folgt erstellt:




```
void setup() {  
  size(200, 200);  
}  
  
void draw() {  
  fill(255, 0, 0);  
  rect(50, 50, 100, 100);  
}
```

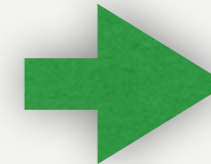
  
Compile & Run



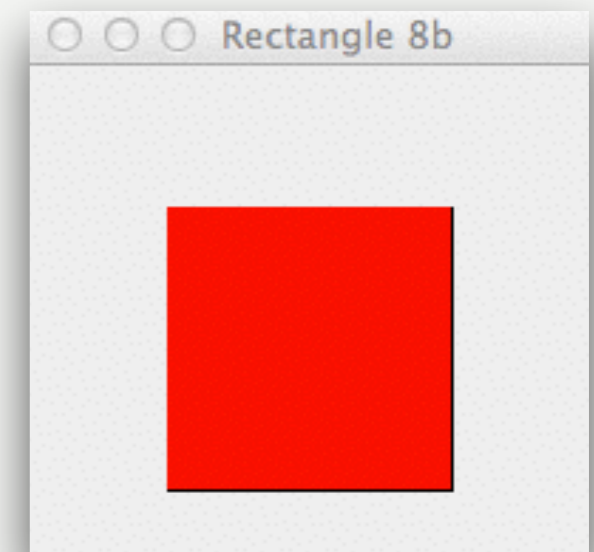


- In Java wird das gleiche Ergebnis mit etwas mehr Aufwand wie folgt erreicht:

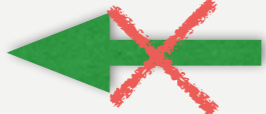

```
Rectangle.java 
1 import java.awt.Color;
2 import java.awt.Graphics;
3
4 import javax.swing.JComponent;
5 import javax.swing.JFrame;
6
7 class MyCanvas extends JComponent {
8
9     public void paint(Graphics g) {
10         g.drawRect(50, 50, 100, 100);
11         g.setColor(Color.RED);
12         g.fillRect(50, 50, 100, 100);
13     }
14 }
15
16
17 public class Rectangle {
18     public static void main(String[] a) {
19         JFrame window = new JFrame();
20         window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         window.setSize(200, 200);
22         window.getContentPane().add(new MyCanvas());
23         window.setTitle("Rectangle 8b");
24         window.setVisible(true);
25     }
26 }
27 }
```



Compile & Run



- Java ist aufwendiger als Processing, aber dafür gewinnen wir deutlich mehr Flexibilität und mehr Möglichkeiten
- Java ist plattformunabhängig, ebenso wie Processing
- Java bildet die Grundlage für sehr viele moderne Programmiersprachen (JavaScript, Processing usw.)
- Java ist *die* Programmiersprache im akademischen Umfeld und sehr verbreitet in der Industrie. Demnach: Viele (gute) Lehrbücher, Anleitungen und Tutorials im Internet
- Sehr guter Toolsupport (Hilfssoftware) für die Entwicklung von Java
- Große Anzahl an modernen Bibliotheken und Erweiterungen

- Processing enthält bereits eine Entwicklungsumgebung (Integrated Development Environment, IDE) bei der Installation
- Für Java gibt es viele IDEs. Die verbreitetsten sind
  - Eclipse 
  - NetBeans
  - IntelliJ 
- Wir verwenden IntelliJ (IntelliJ 2017 (Community Edition))

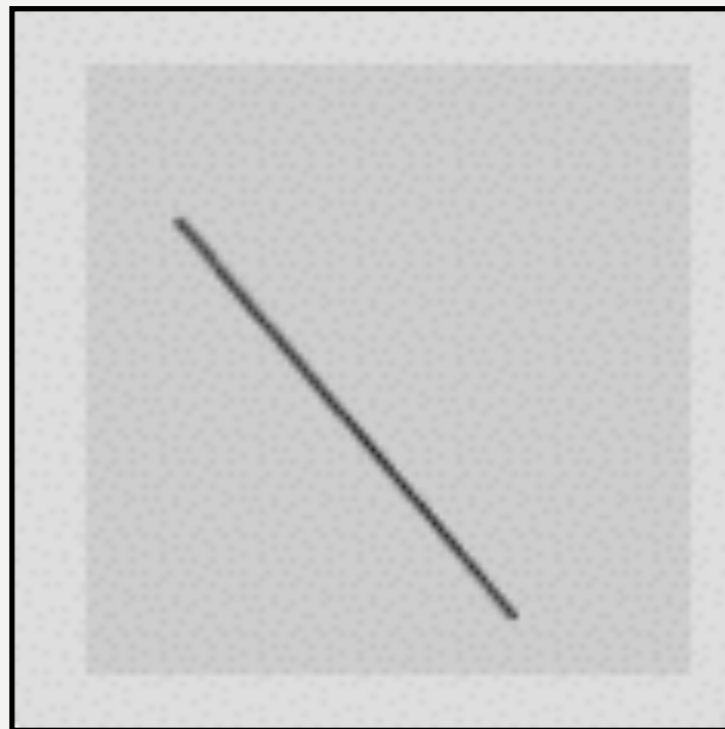
# Anwendung



- Anleitungen zur Installation von IntelliJ / Java findet Ihr viele im Internet. Eine (unverbindliche) Auswahl gibt es hier:
- <https://www.jetbrains.com/help/idea/2017.1/requirements-for-intellij-idea.html>  
IntelliJ Installation für alle Betriebssysteme (englisch)
- <https://www.youtube.com/watch?v=qscQ3XT6qu4>  
IntelliJ Installation und „HelloWorld“ für Windows

- Alle in der Vorlesung programmierten Beispiele werden Euch nach der Vorlesung zur Verfügung gestellt
- Versucht aufzupassen und/oder parallel selbst zu programmieren
- Kommentare und Erklärungen sind im Quellcode enthalten

- In **Processing** wurden Variablen wie folgt definiert:
- **int** begin\_x = 15; //Integer-Variable  
**int** begin\_y = 25;  
**int** end\_x = 70;  
**int** end\_y = 90;
- Ausgabe: *line(begin\_x, begin\_y, end\_x, end\_y);*



- In **Java** sieht es ganz ähnlich aus:
- **int** begin\_x = 15; //Integer-Variable  
**int** begin\_y = 25;  
**int** end\_x = 70;  
**int** end\_y = 90;
- Ausgabe: *Graphics.drawLine(20, 100, 120, 100);*





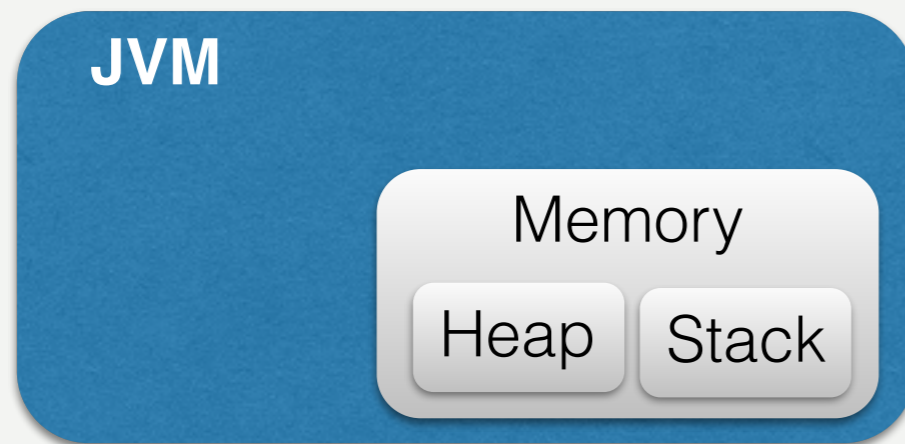
- Ganze Zahlen

Name	Größe im Speicher	Zahlenbereich
byte	1 Byte	-128 bis 127 ( $2^8$ )
short	2 Byte	-32768 bis 32767 ( $2^{16}$ )
int	4 Byte	2.147.483.648 - 2.147.483.648 ( $2^{32}$ )
long	8 Byte	... $2^{64}$

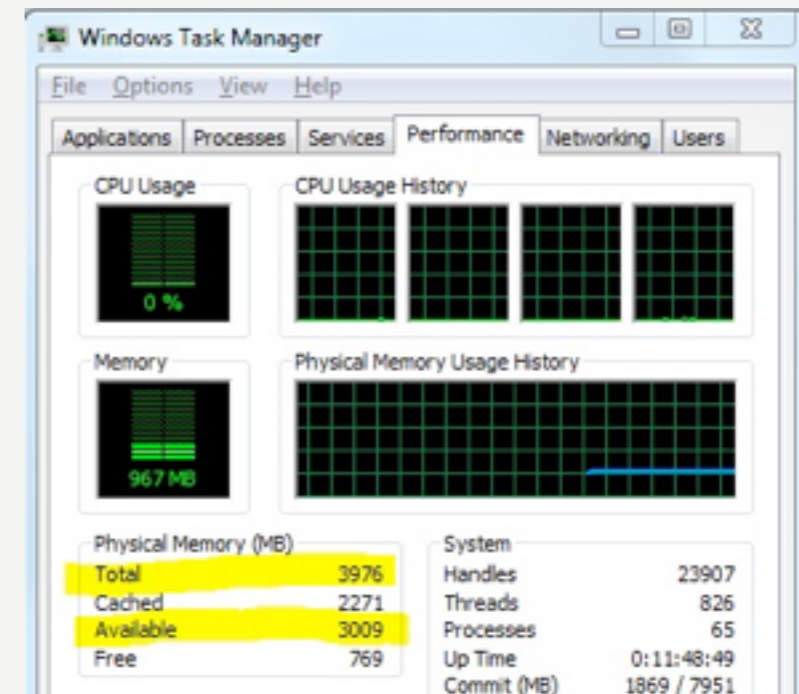
- Gleitkommazahlen

Name	Größe im Speicher	Zahlenbereich
float	4 Byte	Genauigkeit: 7-8 Dezimalstellen
double	8 Byte	Genauigkeit: 15-16 Dezimalstellen

- Java verwaltet seine Variablen in zwei Speicherbereichen
  - **Stack** (Einfache Datentypen wie int, float)
  - **Heap** (Komplexe Datentypen wie Objekte)



bspw. „Java.exe“



Windows Speichermanagement

- Der besondere Datentyp String (Zeichenketten)
- Strings sind Zeichenketten, die aus Buchstaben, Zahlen und Sonderzeichen bestehen können
- Darstellung durch Anführungszeichen
- Beispiel: `String myString = "Java is so cool!";`  
Ausgabe: Java is so cool!
- Besonderheit: Zeilenumbruch mit „`\n`“  
`String myString = "Java is \n so cool!";`  
Ausgabe: Java is  
so cool!



- **Zuweisungen und Operationen** (ähnlich zu Processing)

```
1 public class General {
2
3     public static void main(String[] args) {
4         double d = 3.14; // Achtung! Komma ist der Punkt
5         int zahl_1 = 30;
6         int zahl_2 = 50;
7         int ergebnis = zahl_1 + zahl_2;
8         System.out.println("Das Ergebnis lautet: " + ergebnis);
9
10        String myString = "Hi there!";
11        System.out.println(myString);
12
13    }
14
15 }
16
```



- **If / else**

```
1 public class Kontrollstrukturen_If {
2
3     public static void main(String[] args) {
4         int age = 16;
5         if (age < 18) { //Wenn Alter unter 18 Jahren
6             System.out.println("Ey! Du kommst hier net rein!");
7         } else { //Wenn Alter über 18 Jahren
8             System.out.println("Willkommen! :-)");
9         }
10
11     }
12
13 }
```



- **While und For-Schleifen**

```
1 public class Loop {
2     public static void main(String[] args) {
3         boolean isInteresting = true;
4         while (isInteresting) { // solange es interessant ist
5             System.out.println("That's so interesting!");
6         }
7         for (int i = 0; i < 10; i++) { // zehn mal durchlaufen
8             System.out.println("Aktueller Schleifendurchlauf:" + i);
9         }
10    }
11 }
12 }
```

# Verknüpfung



Quiztime :-)

A:

B:

C:

D:



*MSMJOKRQ*



## Quiz: Was ist IntelliJ und wofür verwenden wir es?

A: IntelliJ ist auch eine Programmiersprache und wir verwenden es nicht

B: IntelliJ ist eine Entwicklungsumgebung, mit der wir Java programmieren

C: So etwas gibt es nicht!

D: IntelliJ ist die Laufzeitumgebung, ohne die Java nicht funktionieren würde

## Quiz: Was ist IntelliJ und wofür verwenden wir es?

A: IntelliJ ist auch eine Programmiersprache und wir verwenden es nicht

B: IntelliJ ist eine Entwicklungsumgebung, mit der wir Java programmieren

C: So etwas gibt es nicht!

D: IntelliJ ist die Laufzeitumgebung, ohne die Java nicht funktionieren würde



## Quiz: Welche der folgenden Aussagen über Java und Processing sind richtig?

A: Java ist eine Programmiersprache und Processing nicht

B: Beides sind Programmiersprachen, jedoch ist Java komplexer

C: Java und Processing sind das gleiche

D: Processing ist eine Verbesserung/Weiterentwicklung von Java

## Quiz: Welche der folgenden Aussagen über Java und Processing sind richtig?

A: Java ist eine Programmiersprache und Processing nicht

B: Beides sind Programmiersprachen, jedoch ist Java komplexer

C: Java und Processing sind das gleiche

D: Processing ist eine Verbesserung/Weiterentwicklung von Java



## Quiz: Wie oft wird die folgende Schleife durchlaufen?

```
for (int i = 0; i <= 15; i++) { // ??? mal durchlaufen  
    System.out.println("Aktueller Schleifendurchlauf:" + i);  
}
```

A: Kein mal

B: 15 mal

C: 16 mal

D: 17 mal

## Quiz: Wie oft wird die folgende Schleife durchlaufen?

```
for (int i = 0; i <= 15; i++) { // ??? mal durchlaufen  
    System.out.println("Aktueller Schleifendurchlauf:" + i);  
}
```

A: Kein mal

B: 15 mal

C: 16 mal

D: 17 mal

# Ausblick

- Die kommenden Übungen dienen als “Beratungsstunde”
- Besucht die Übungen und bearbeitet die Übungsaufgaben dort
- Der Verzug zwischen Vorlesung und Übung wird damit reduziert



## Nächstes Übungsblatt

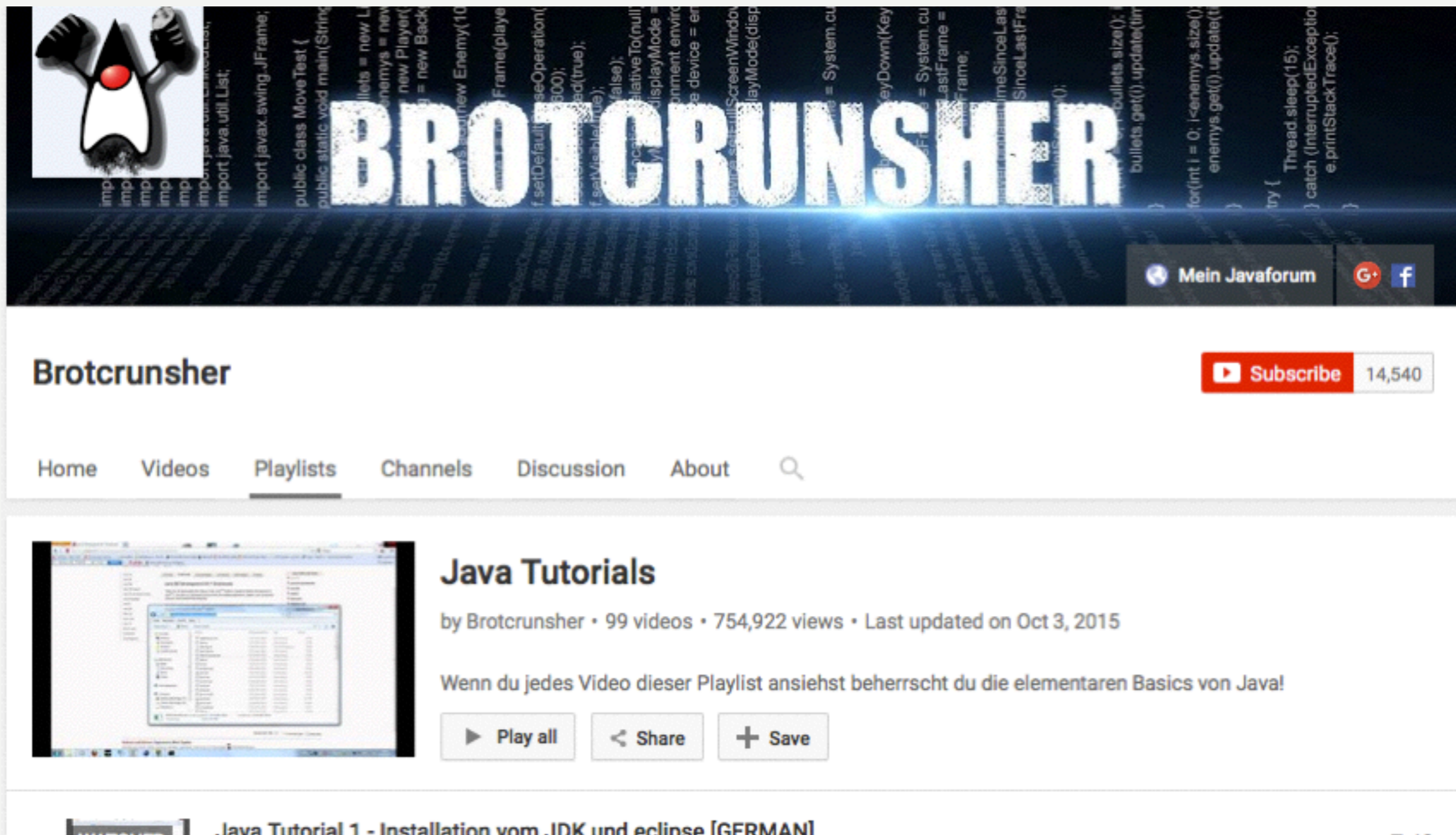


- Das nächste Übungsblatt enthält Aufgaben zum Einstieg in Java
- Löst diese Aufgaben im Team



- Nach einer Eingewöhnungsphase in Java, werdet ihr in den Übungen Stück für Stück das Spiel „Pong“ entwickeln
- Komplexe Codefragmente werden von uns zur Verfügung gestellt
- Nach der letzten Übungsstunden wird die “Musterlösung” bekannt veröffentlicht
- Für alle, die „Pong“ nicht kennen:  
<https://www.youtube.com/watch?v=ZmrJ30BmfBM>

- Eine Empfehlung eurer Vorgänger



The image shows a screenshot of a YouTube channel page for 'Brotcrunsher'. The channel banner features a cartoon character with a red eye and a black body, set against a background of Java code. The channel name 'Brotcrunsher' is prominently displayed, along with a 'Subscribe' button and a subscriber count of 14,540. Below the channel name, there are navigation tabs for Home, Videos, Playlists, Channels, Discussion, and About. A featured playlist titled 'Java Tutorials' is shown, with a video thumbnail depicting a code editor. The playlist description reads: 'Wenn du jedes Video dieser Playlist ansiehst beherrscht du die elementaren Basics von Java!'. Below the description are buttons for 'Play all', 'Share', and 'Save'. At the bottom of the screenshot, the title of the first video in the playlist is visible: 'Java Tutorial 1 - Installation vom JDK und eclipse [GERMAN]'.

<https://www.youtube.com/playlist?list=PL71C6DFDDEF73835C2>

- Bevor Ihr IntelliJ verwenden könnt, *müsst* Ihr Euch die Java Laufzeitumgebung für Entwickler installieren (Java JDK; Java Development Kit)
- <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Verwendet dabei die zu diesem Zeitpunkt aktuelle Version 8u151

# Fragen?

Vielen Dank für  
Eure Zeit!