

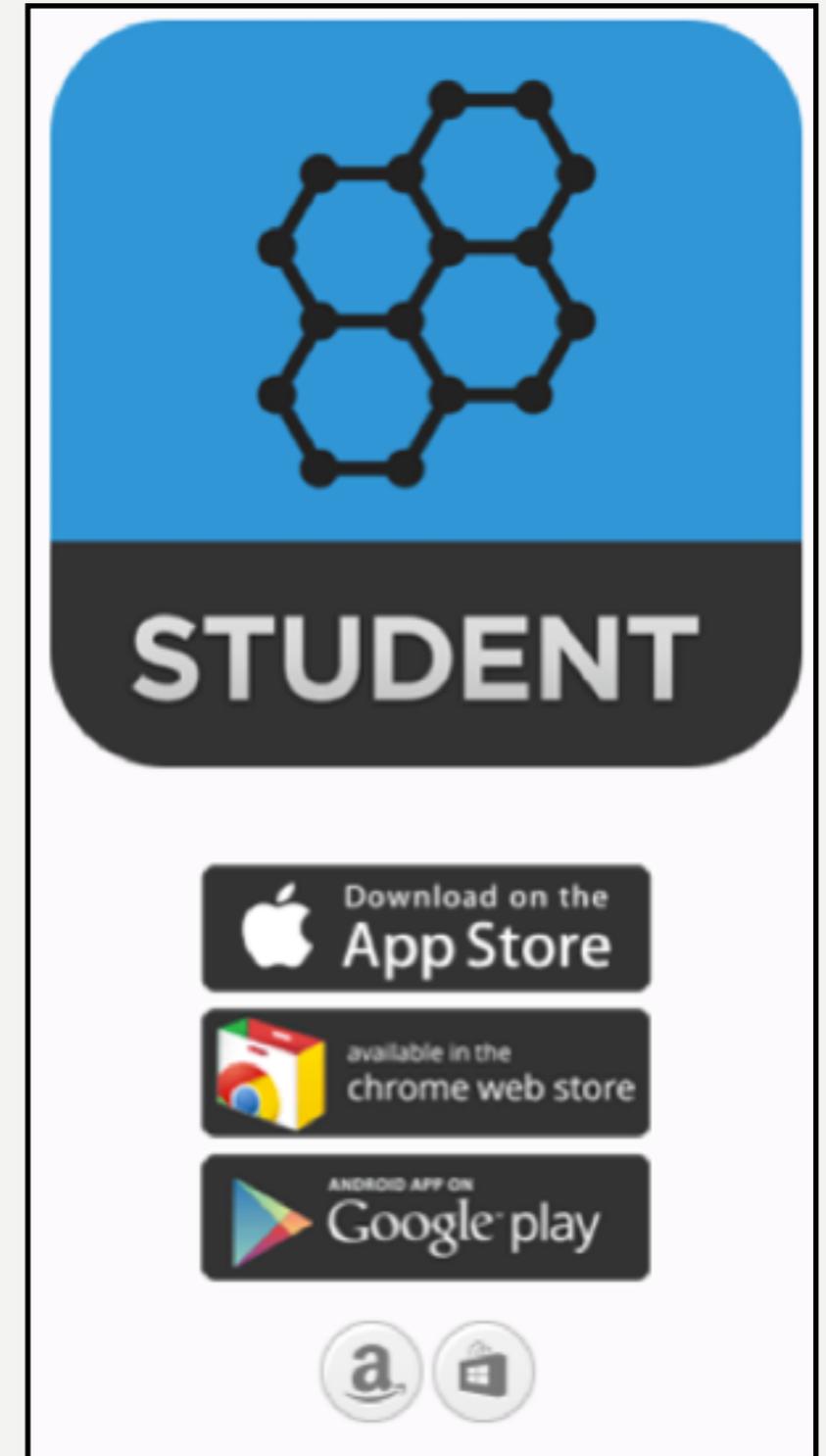
Einführung in die Programmierung für Nebenfach Medieninformatik

Beat Rossmly, 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



Objekte und Methoden

Verwendung von IntelliJ, Klassen, Konzepte,
Objekte, Verhalten von Objekten bei
Java Swing, ActionListener

Beat Rossmly, Michael Kirsch

Verspätete Kurzvorstellung :-)

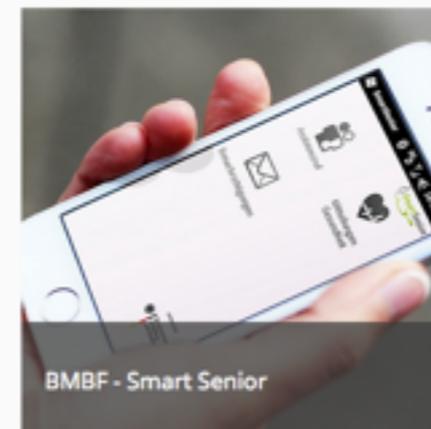
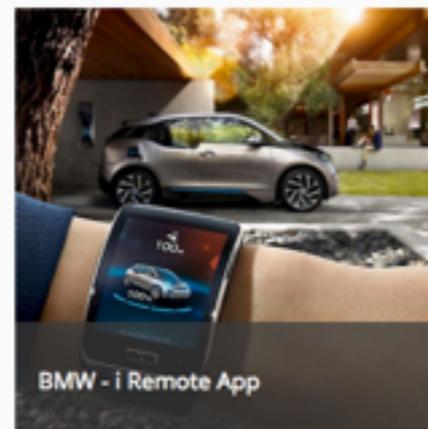
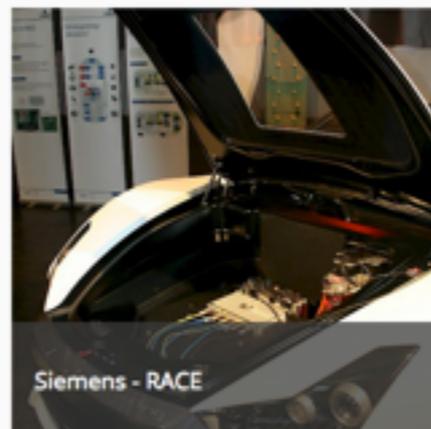
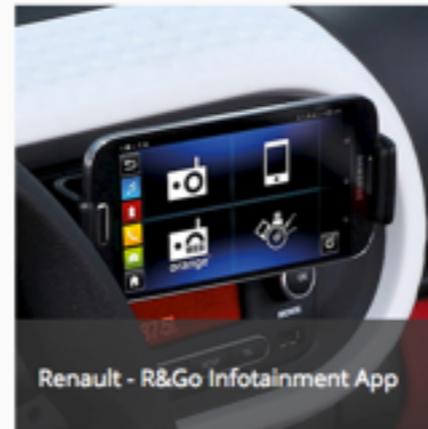
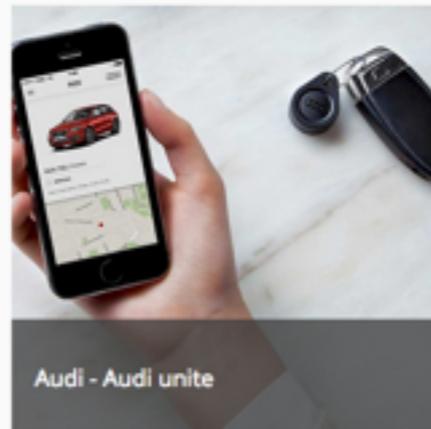
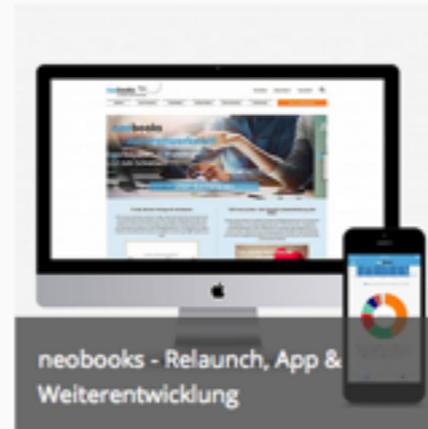
Wer bin ich?



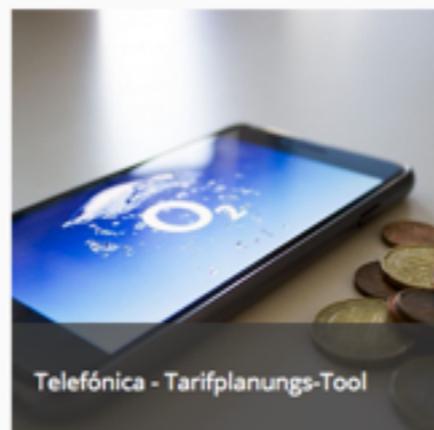
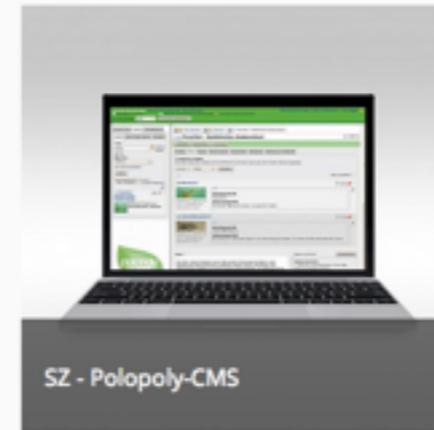
- Michael Kirsch (29 Jahre)
- Berufsausbildung zum Fachinformatiker (Anwendungsentwicklung)
- B.Sc. Hochschule Hof (Technische Informatik)
- Auslandsaufenthalt am International Institute of Information Technology Bangalore / Indien (M.Eng.)
- M.Sc. LMU München (Informatik)
- Softwarearchitekt / Softwareentwickler bei  **jambit**
WHERE INNOVATION WORKS



Innovationstories



Innovationstories



Jambit - Where Innovation Works



Kaffeestory

Ende 1999 saßen die beiden Geschäftsführer Peter Fellingner und Markus Hartinger bei einer Tasse Cappuccino in ihrem Lieblingscafé "kaffee & mehr" am Viktualienmarkt zusammen. Auf der Suche nach einem geeigneten Namen für ihre neue Firma blätterten sie in einem Kaffeehandbuch. Darin tauchte der Name "jambit" auf: Es sei die beste Kaffee-Plantage auf der indonesischen Insel Java, hieß es. Der Name enthielt "IT", in Suchmaschinen gab es dazu keine Einträge, Domains waren frei – beschlossene Sache!

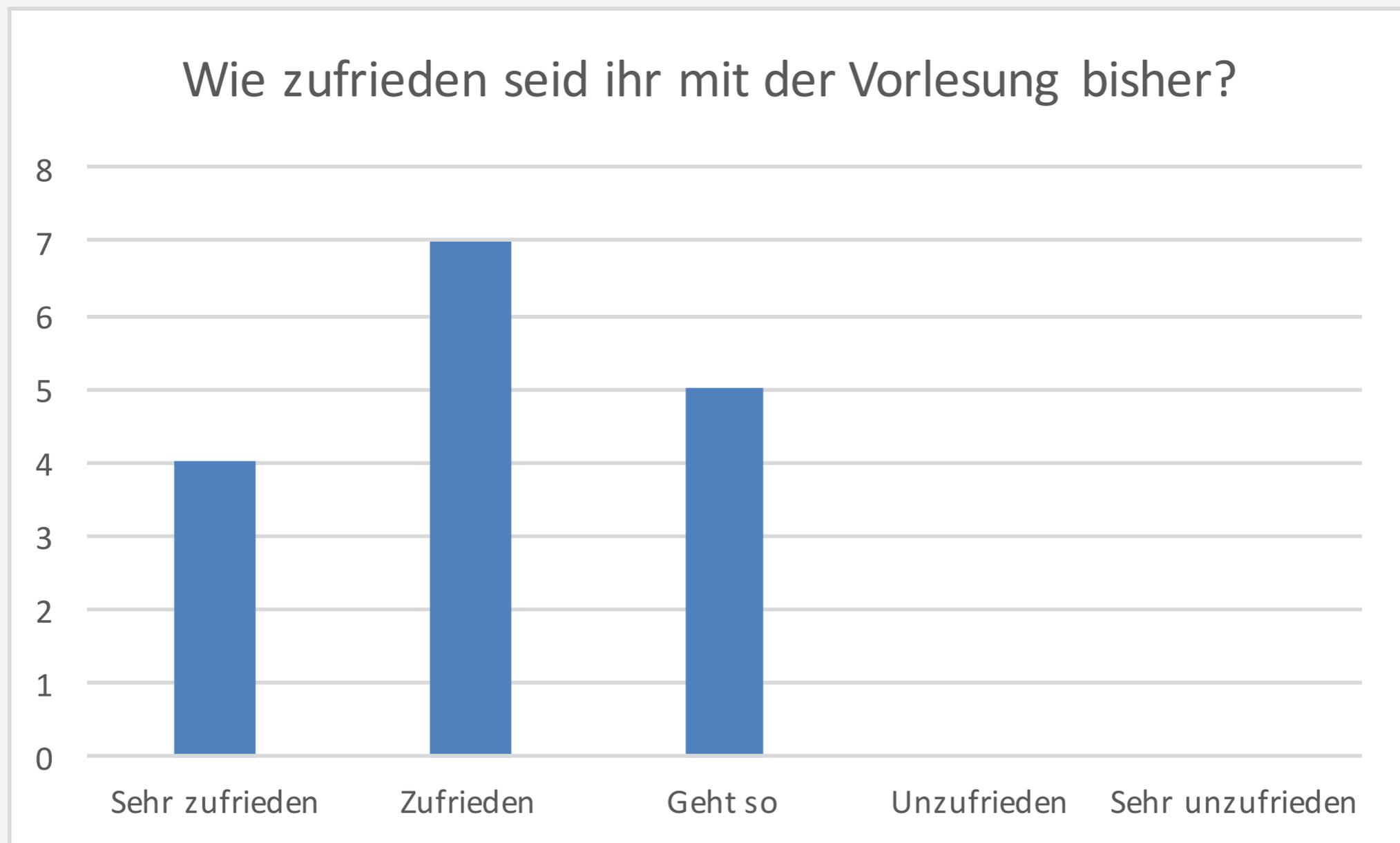
Natürlich war die Versuchung groß, genau diesen legendären Kaffee auch tatsächlich probieren zu können. Die Suche nach der exakten Lage der Plantage war nicht einfach, und erst der Hinweis in einem Reiseführer über Indonesien, dass auch "Jampit" oder gar "Djampit" phonetisch korrekte Transkriptionen aus der indonesischen Sprache sind, führte zu der sagenumwobenen Hochebene auf dem Vulkangestein Javas.

Seitdem lassen wir regelmäßig originalen, unvermischten Kaffee aus Jampit kommen und in München speziell für uns frisch von Hand rösten.

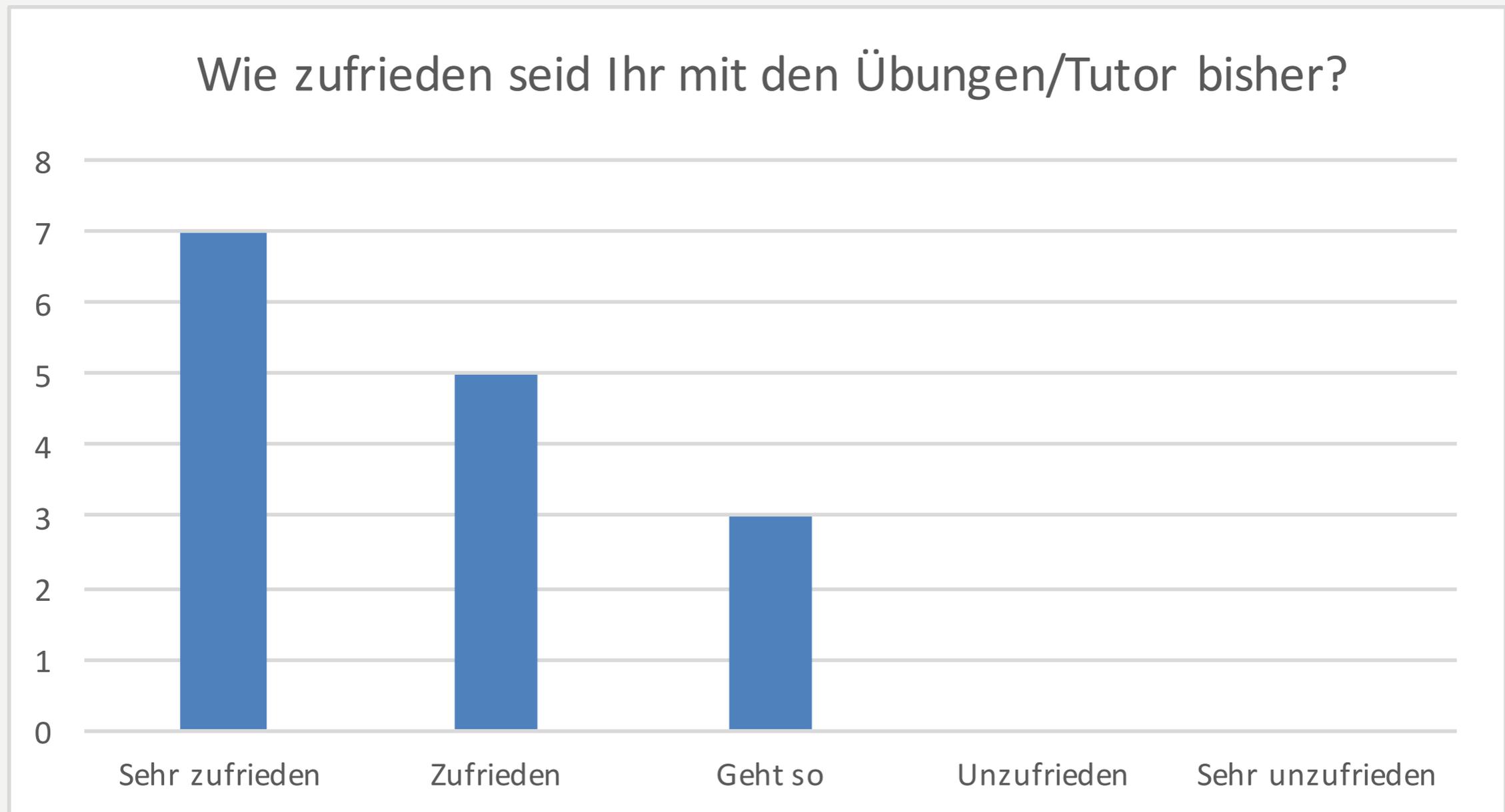


Ergebnisse der Umfrage

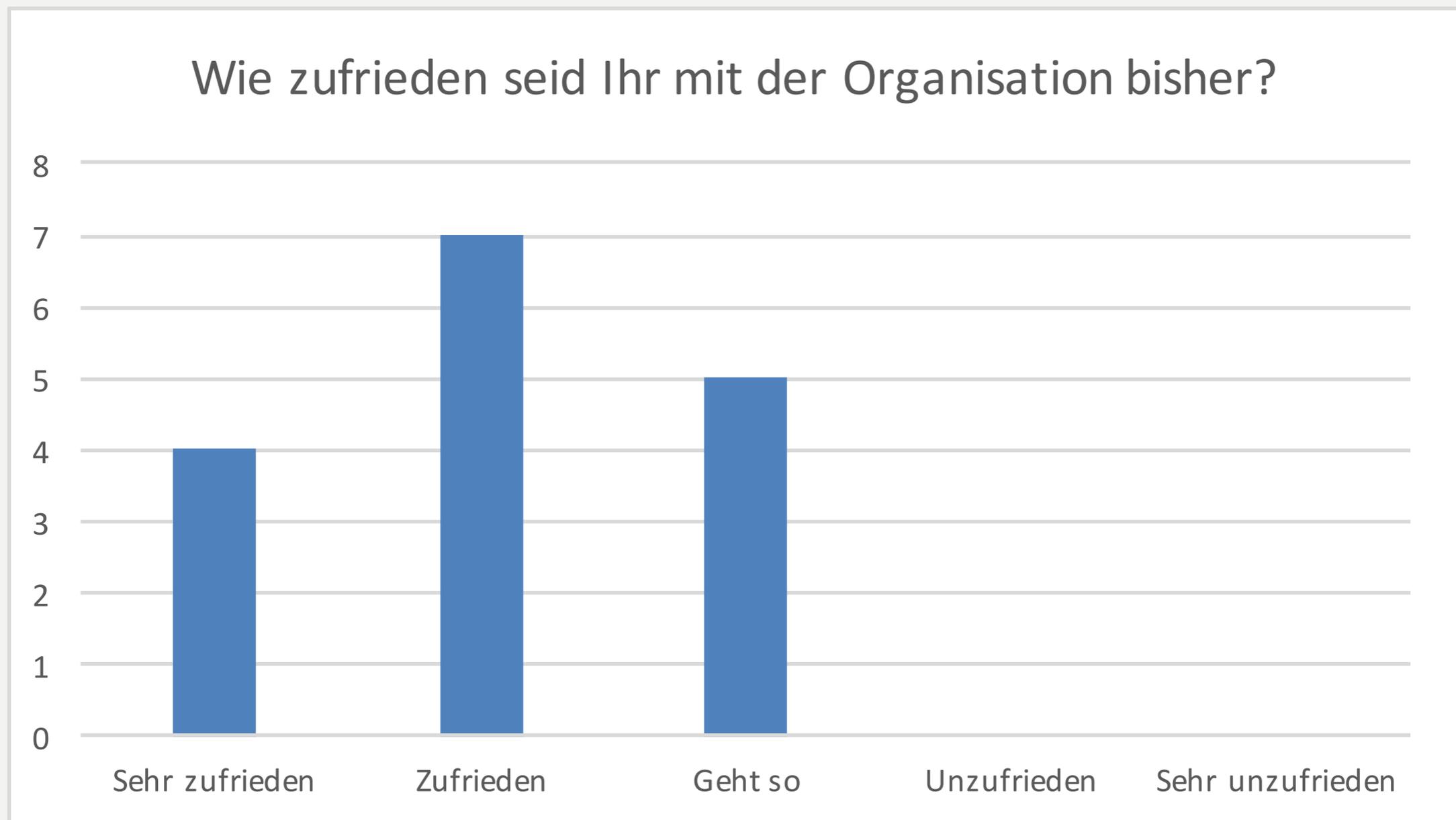
Ergebnis der Umfrage zur Vorlesungszufriedenheit



Ergebnis der Umfrage zur Vorlesungszufriedenheit



Ergebnis der Umfrage zur Vorlesungszufriedenheit



Quiztime :)



Quiz: Welche Ausgabe erzeugt das folgende Programm?



```
public class Quiz01 {  
  
    public static void main(String[] args) {  
  
        int result;  
        result = add( zahl1: 3, zahl2: 4);  
        System.out.println("Das Ergebnis lautet: " + result);  
  
        System.out.println("Das Ergebnis lautet: " + addiere( ersteZahl: 3, zweiteZahl: 4));  
    }  
  
    public static int add(int zahl1, int zahl2) {  
        int ergebnis;  
        ergebnis = zahl1 + zahl2;  
  
        return ergebnis;  
    }  
  
    public static int addiere(int ersteZahl, int zweiteZahl) {  
        return ersteZahl + zweiteZahl;  
    }  
}
```



Quiz: Welche Ausgabe erzeugt das folgende Programm?

A: Das Ergebnis lautet: 7
Das Ergebnis lautet: 7

B: Das Ergebnis lautet: 7 Das Ergebnis lautet: 7

C: Das Ergebnis lautet: 7
Das Ergebnis lautet: 14

D: Das Ergebnis lautet: 14
Das Ergebnis lautet: 7

Quiz: Welche Ausgabe erzeugt das folgende Programm?

A: Das Ergebnis lautet: 7
Das Ergebnis lautet: 7

B: Das Ergebnis lautet: 7 Das Ergebnis lautet: 7

C: Das Ergebnis lautet: 7
Das Ergebnis lautet: 14

D: Das Ergebnis lautet: 14
Das Ergebnis lautet: 7

Quiz: Welche Ausgabe erzeugt das folgende Programm?

```
public class Quiz02 {  
    public static void main(String[] args) {  
        int i, h;  
  
        i = 4;  
        h = i;  
  
        if (i == h) {  
            int k = 42;  
            int a = 100000;  
            h = k + i;  
        }  
        i = 0;  
        h = i;  
  
        System.out.println("i + a");  
    }  
}
```



Quiz: Welche Ausgabe erzeugt das folgende Programm?

A: 0

B: Das Programm funktioniert so nicht!

C: i + a

D: 100004



Quiz: Welche Ausgabe erzeugt das folgende Programm?

A: 0

B: Das Programm funktioniert so nicht!

C: $i + a$

D: 100004



Quiz: Was wird benötigt, um Java zu programmieren UND das Programm auszuführen?

A: Gute Drogen und viel Hoffnung

B: Die JRE (Java Runtime Environment) reicht aus

C: Das neue Macbook mit Touch Bar, 16 GB Arbeitsspeicher für 3199,00€

D: JDK (Java Development Kit) und JRE (Java Runtime Environment)

Quiz: Was wird benötigt, um Java zu programmieren UND das Programm auszuführen?

A: Gute Drogen und viel Hoffnung

B: Die JRE (Java Runtime Environment) reicht aus

C: Das neue Macbook mit Touch Bar, 16 GB Arbeitsspeicher für 3199,00€

D: JDK (Java Development Kit) und JRE (Java Runtime Environment)

I. Einleitung

1. Wie geht es weiter?
2. Java Swing

II. Theorie

1. GUI in Java
2. Aufbau von Swing
3. JFrame
4. JPanel
5. ActionListener

III. Anwendung

1. Java Swing in Action

IV. Verknüpfung

1. Beginn Eures Projekts

V. Ausblick

1. Nächste Vorlesung
2. Übung

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

Swing

Objekte/Klassen

...



Einleitung

Wie geht es weiter?



- Einführung in Java Swing (Oberflächenprogrammierung)
- Alle folgenden Vorlesungen bauen auf Eurem Projekt auf und es werden Stück für Stück neue Techniken vermittelt, damit Ihr Euer Spiel entwickeln könnt
- Dieses Mal: Einfache Oberflächen, Grafiken und ActionListener

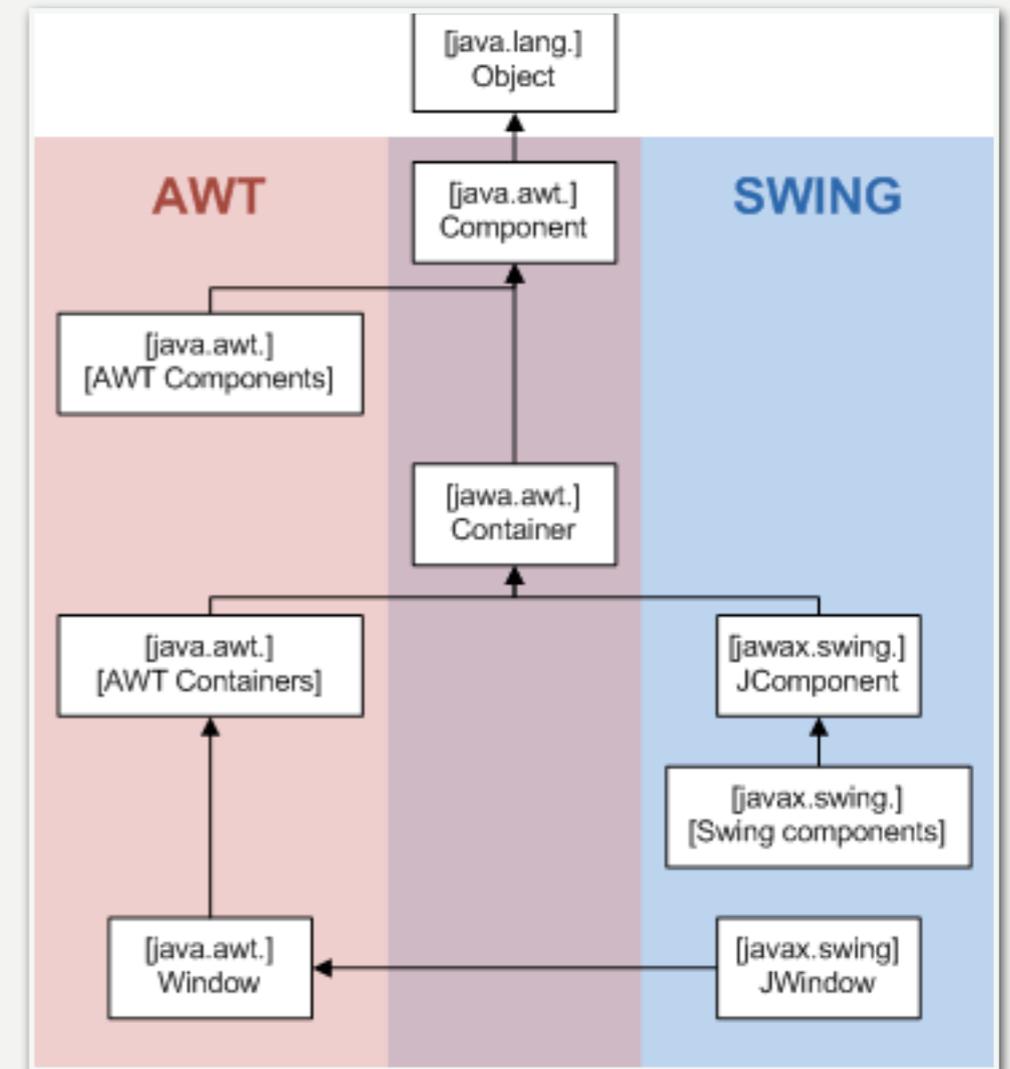
Prozedural (Anfänge der Programmierung)

- Fest definierte Reihenfolge des Ablaufs
- Sequentielle Abarbeitung
- Verwendung von einfachen Datentypen (bspw. int, double)

Objektorientiert (seit den 90er Jahren)

- Reihenfolge abhängig von Verwendung der Objekte
- Objekten werden Eigenschaften und Verhalten zugewiesen
- Definition von eigenen Datentypen bzw. Klassen (bspw. Adressbucheintrag oder Klasse *Mensch*)

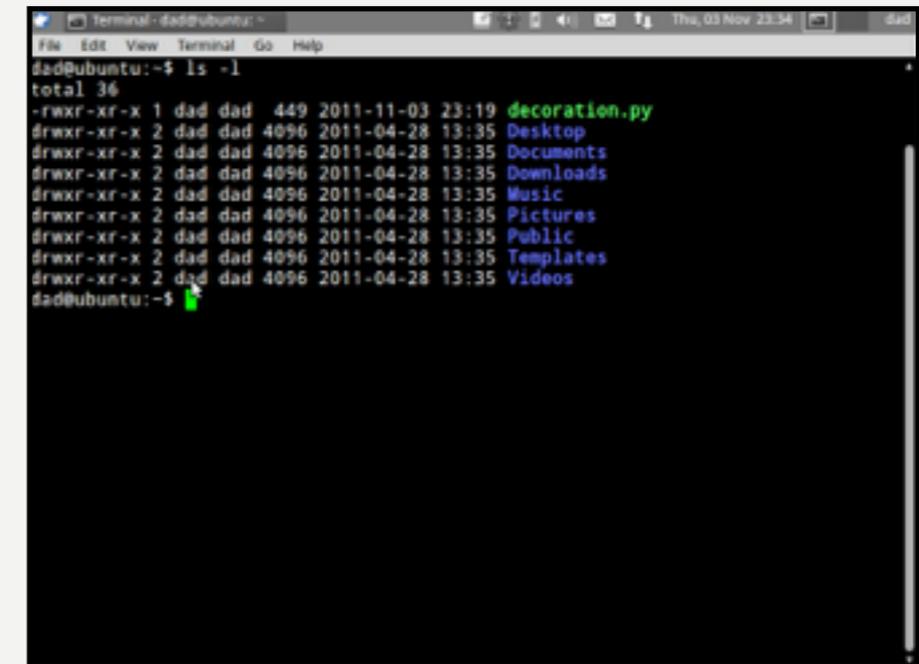
- Mit *Swing* werden Oberflächen oder auch GUI (Graphical User Interfaces) entwickelt
- Es besteht u.a. aus vielen verschiedenen Komponenten (JFrame, JPanel, JButton, JTextfield, ...)
- **Diese Komponenten sind Objekte!**
- Viele Beispiele unter <https://docs.oracle.com/javase/tutorial/uiswing/examples/components/>
- Aufbau (siehe Abbildung)



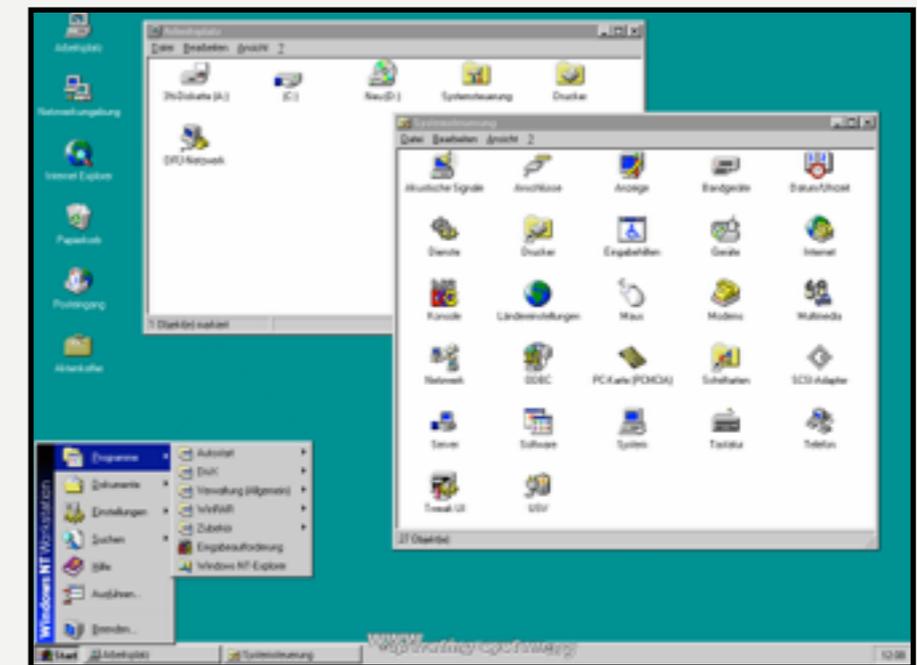


Theorie

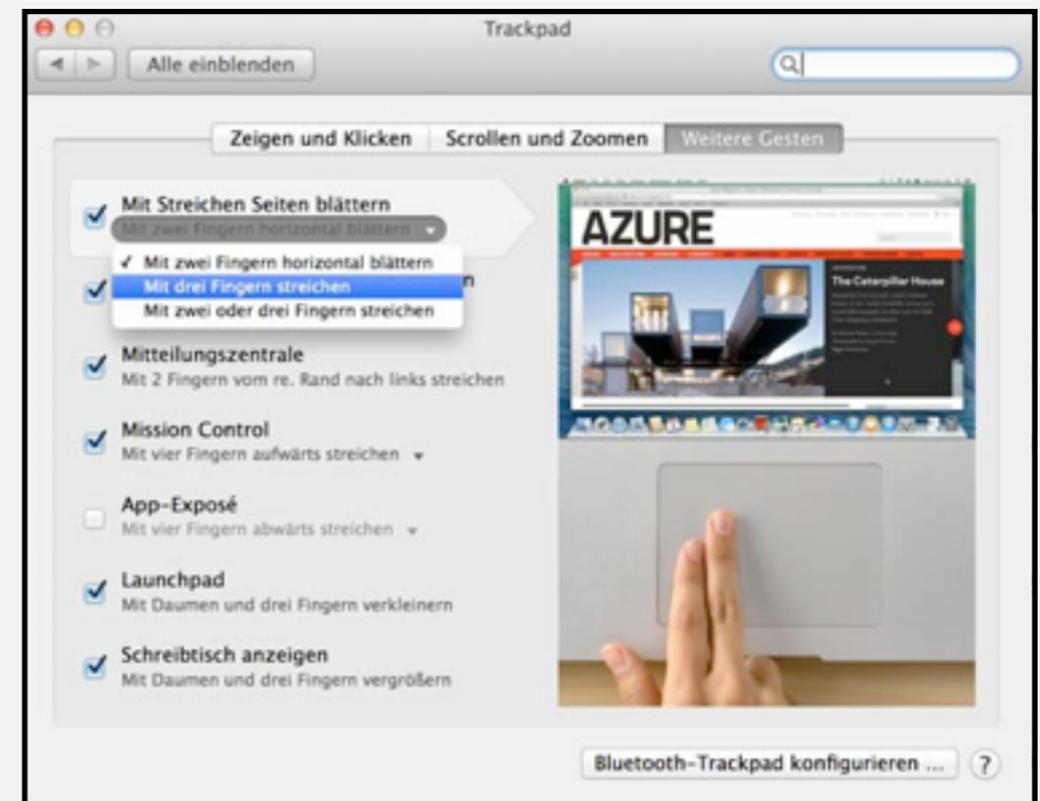
- Am Anfang gab es Terminals
- Anschließend kamen mehr und mehr grafische Anwendungen auf
- Anfangs noch einfach und funktional (Maus- & Tastatursteuerung)

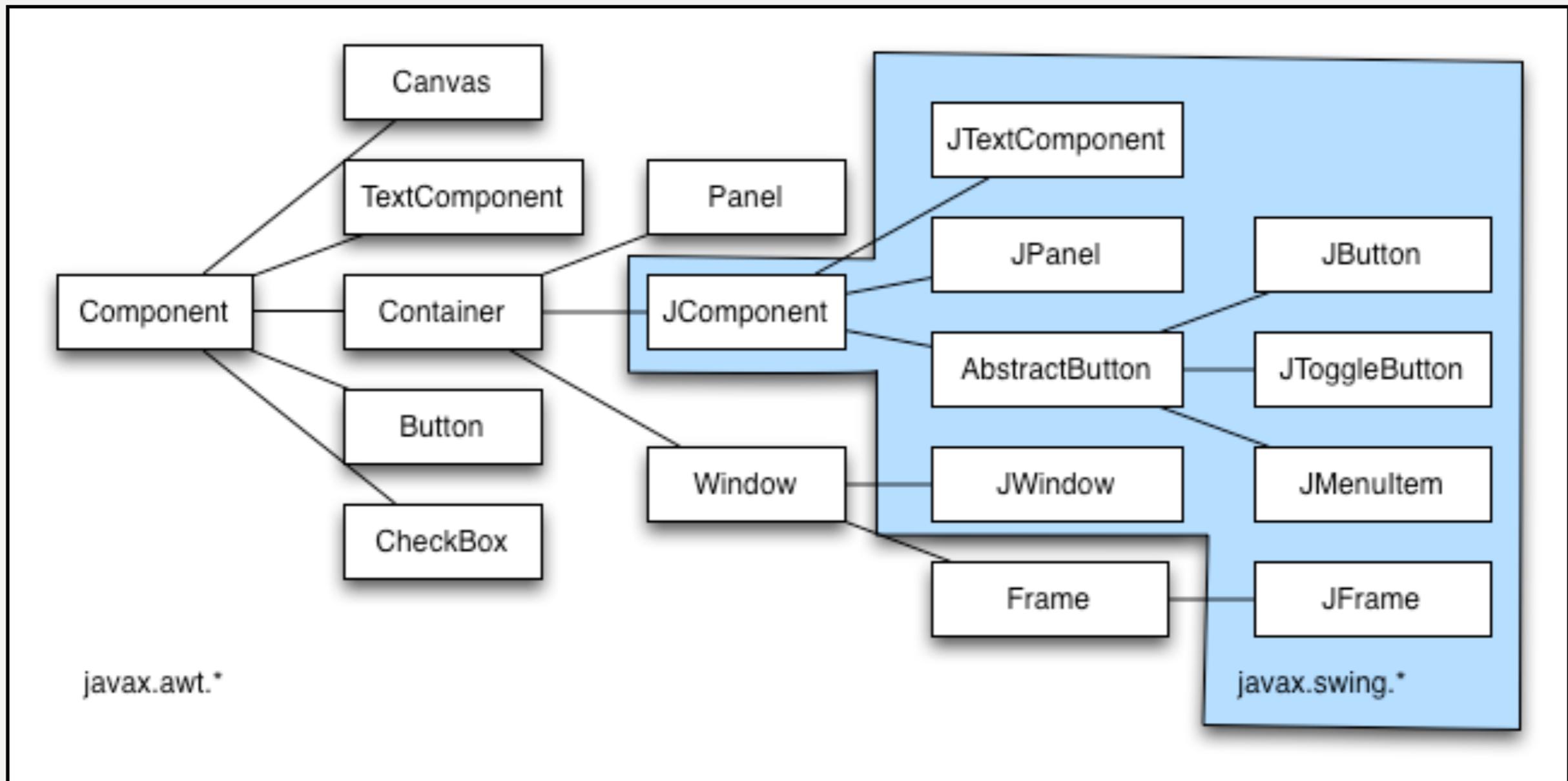


```
Terminal - dad@ubuntu:~  
File Edit View Terminal Go Help  
dad@ubuntu:~$ ls -l  
total 36  
-rwxr-xr-x 1 dad dad 449 2011-11-03 23:19 decoration.py  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Desktop  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Documents  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Downloads  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Music  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Pictures  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Public  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Templates  
drwxr-xr-x 2 dad dad 4096 2011-04-28 13:35 Videos  
dad@ubuntu:~$
```

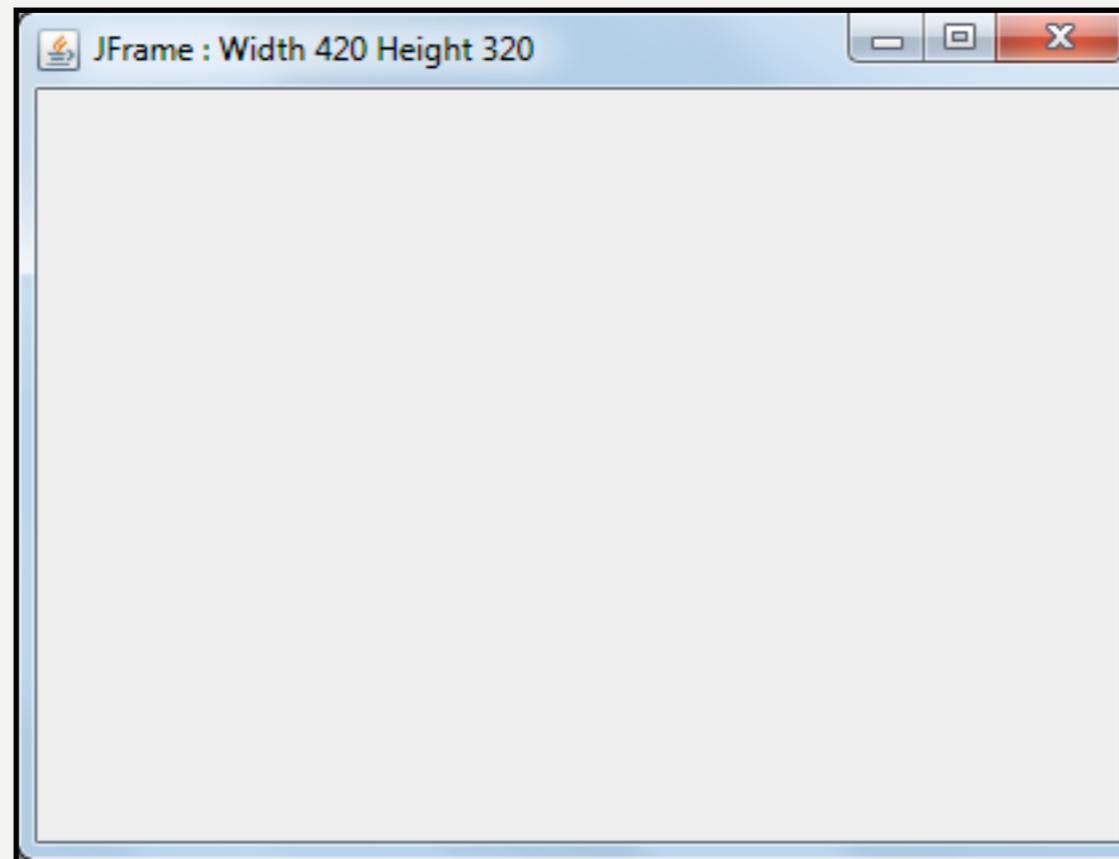


- Später komplexer und schicker (Einfache und Multitouch-Gesten)
- In Java ist das alles ebenfalls möglich
- Bibliothek in Java zur GUI-Programmierung: AWT bzw. SWING
- Swing ist wie Lego: Es werden Komponenten ineinander „gesteckt“
- AWT wurde später durch SWING abgelöst, aber manche Komponenten von AWT sind in SWING verfügbar

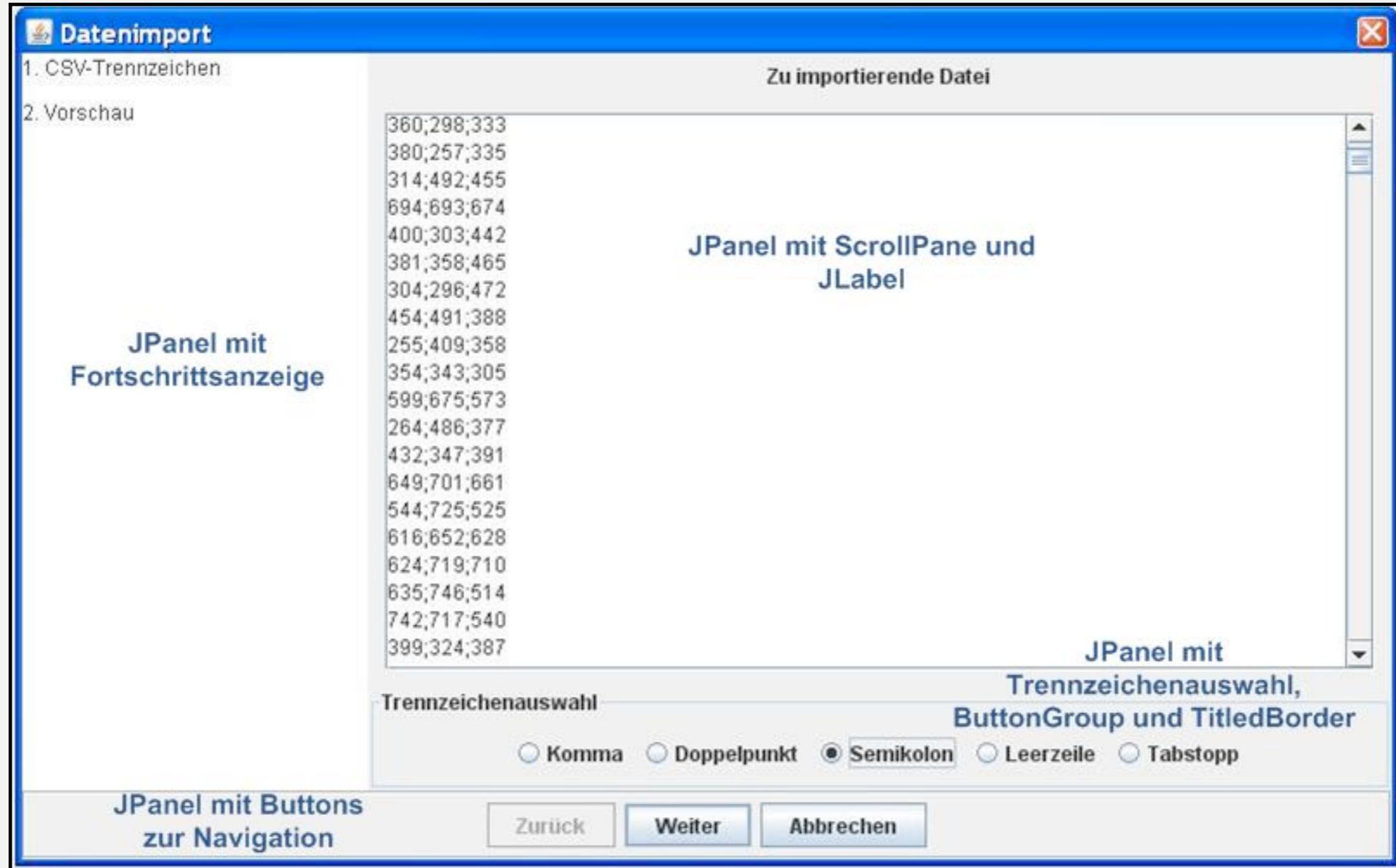




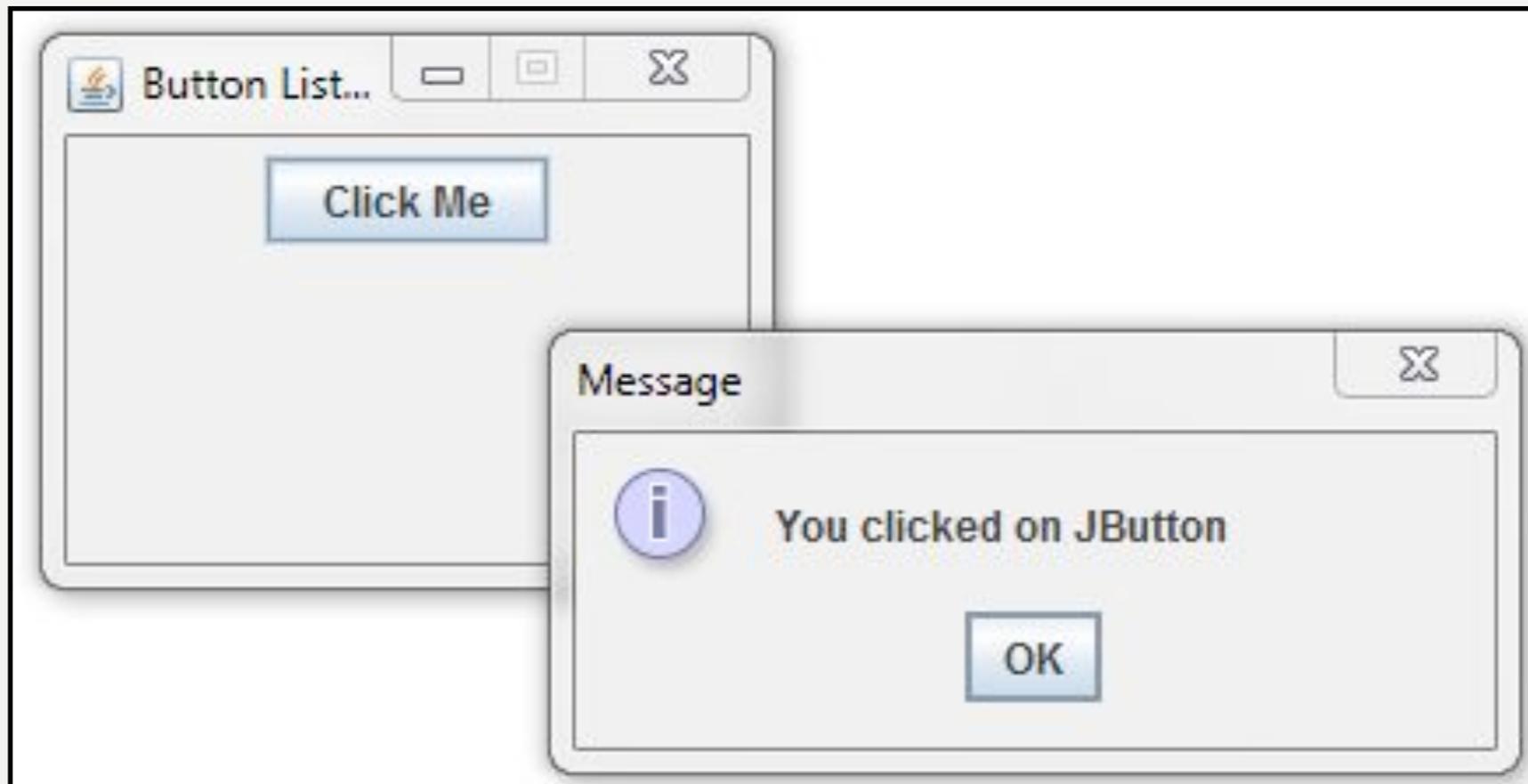
Vererbungshierarchie in Java (alles von Component abgeleitet)



Einfacher JFrame



Ein JFrame enthält mehrere JPanels



ActionListener reagieren auf Actions (bspw. Tastendrücke)

Anwendung



- Einfacher JFrame mit der Größe 600x800 Pixel

```
10
11 public class Main {
12
13     public static void main(String[] args) {
14
15         JFrame fenster = new JFrame(); //Erstelle einen neuen JFrame
16         fenster.setSize(new Dimension(600, 800)); //Fenstergröße 600 Pixel X 800 Pixel
17         fenster.setVisible(true); //Macht das Fenster sichtbar
18
19     }
20
21 }
22
```



- JPanel in einem JFrame

```
14 public class Main {
15
16     public static void main(String[] args) {
17
18         JFrame fenster = new JFrame(); //Erstelle einen neuen JFrame
19         JPanel panel = new JPanel(); //Erstelle ein neues JPanel
20
21         panel.setBorder(BorderFactory.createLineBorder(Color.BLACK)); //Füge Panel einen schwarzen Rand hinzu
22         fenster.add(panel); //Füge das Panel dem Fenster hinzu
23
24         fenster.setSize(new Dimension(600, 800)); //Fenstergröße 600 Pixel X 800 Pixel
25         fenster.setVisible(true); //Macht das Fenster sichtbar
26
27     }
28
29 }
30
```

- KeyListener ermöglichen die Abfrage von Tastatureingaben
- Verwendung von KeyListener in zwei Varianten
 - **Implementierung des Interfaces *KeyListener***
 - Implementierung in einer *anonymen inneren Klasse*
- Damit ein Fenster Key-Events abfragen kann, muss es im *Fokus* stehen
`spielfeld.setFocusable(true); //Aktivierung der „Fokusierbarkeit“`
- KeyListener können die folgenden drei Events abfragen



```
public void keyTyped(KeyEvent e)
public void keyPressed(KeyEvent e)
public void keyReleased(KeyEvent e)
```



```
public class Spielfeld extends JPanel implements KeyListener { //Spielfeld ist ein JPanel und implementiert das Interface KeyListener

    public void Spielfeld() { //Konstruktor; Bisher ohne weitere Funktion
    }

    @Override
    public void keyTyped(KeyEvent e) { //Wenn Taste gedrückt und wieder losgelassen wurde
    }

    @Override
    public void keyPressed(KeyEvent e) { //Wenn eine Taste gedrückt
    if (e.getKeyCode() == KeyEvent.VK_LEFT) { //Wenn die Taste == linke Pfeiltaste
        System.out.println("Linke Pfeiltaste");
    }

    else if (e.getKeyCode() == KeyEvent.VK_RIGHT) { //Wenn die Taste == rechte Pfeiltaste
        System.out.println("Rechte Pfeiltaste");
    }

    else if (e.getKeyCode() == KeyEvent.VK_DOWN) { //Wenn die Taste == untere Pfeiltaste
        System.out.println("Untere Pfeiltaste");
    }

    else if (e.getKeyCode() == KeyEvent.VK_UP) { //Wenn die Taste == obere Pfeiltaste
        System.out.println("Obere Pfeiltaste");
    }

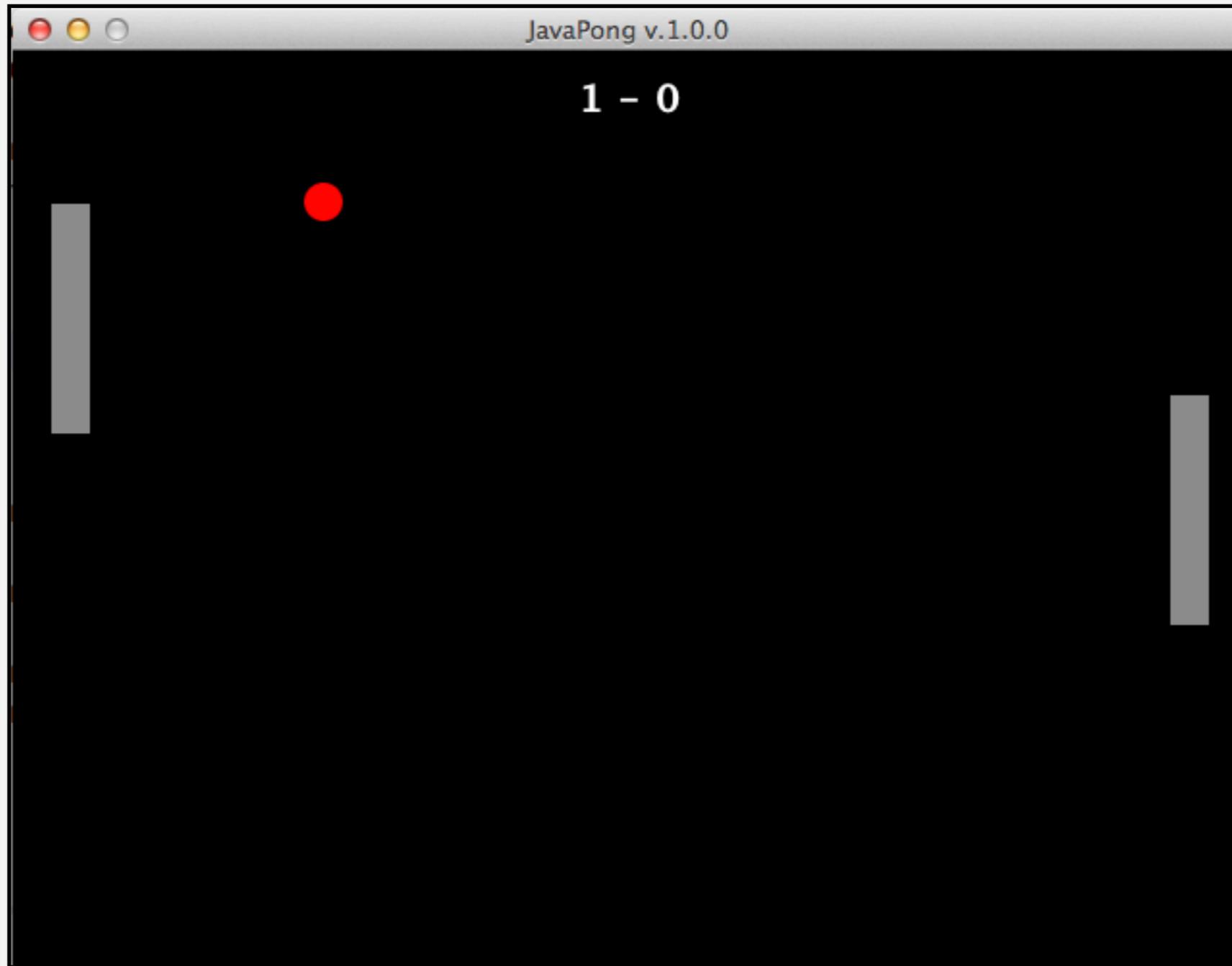
    }

    @Override
    public void keyReleased(KeyEvent e) { //Wenn Taste losgelassen wurde
    }

}
```

Verknüpfung

Pong in Java (JPong)



Ausblick

- In den Übungen werden die Inhalte der heutigen Vorlesung vertieft
 - Selbst Spielfeld erstellen
 - Ball erstellen und bewegen
 - Ball auf Spielfeld halten (Grenzen)
- Nach der letzten Übung, wird eine Musterlösung bereitgestellt
- Nach der Vorlesung wird euch ein “Code-Grundgerüst” bereitgestellt
- **Im neuen Jahr werden wir wir Animationen einführen**

Fragen?



Vielen Dank für
Eure Zeit