

# Session 1: Welcome & Introduction

## Fortgeschrittene Programmierung (Java 2)

Nils Reiter

`nils.reiter@uni-koeln.de`

April 5, 2023

# Section 1

## Introduction

# Contents of this Class

## Advanced programming (in Java)

- ▶ Version control (= git)
- ▶ Recursion
- ▶ Libraries
- ▶ Unit testing
- ▶ Efficient programming
- ▶ Multithreading
- ▶ ...

## Fortgeschrittene Programmierung (Java 2)

- ▶ Teil des Basismoduls “Softwaretechnologie” für Informationsverarbeitung und Medieninformatik
- ▶ Zweiter Teil des Programmierkurses
- ▶ Begleitet von zwei Übungen
  - ▶ Dienstag (10:00-11:30): Dennis Demmer
  - ▶ Donnerstag (10:00-11:30): Theo Bouveyron
  - ▶ Dritter Termin online nachmittags
  - ▶ Anmeldung in Klips
- ▶ Studienleistung: Übungen via GitHub

# Fortgeschrittene Programmierung (Java 2)

- ▶ Teil des Basismoduls “Softwaretechnologie” für Informationsverarbeitung und Medieninformatik
- ▶ Zweiter Teil des Programmierkurses
- ▶ Begleitet von zwei Übungen
  - ▶ Dienstag (10:00-11:30): Dennis Demmer
  - ▶ Donnerstag (10:00-11:30): Theo Bouveyron
  - ▶ Dritter Termin online nachmittags
  - ▶ Anmeldung in Klips
- ▶ Studienleistung: Übungen via GitHub
- ▶ Modulprüfung
  - ▶ Programmierprojekt

Kurswebseite:

<https://uni.koeln/6LDDK>



# Learning Programming

- ▶ Learning to program is hard and takes time
- ▶ It helps to
  - ▶ Regularly do it
  - ▶ Talk about it
  - ▶ Be stubborn
  - ▶ Think formalistic
  - ▶ Be fearless and disrespectful
  - ▶ Read documentation
  - ▶ Try to understand your mistakes
- ▶ It's ok to make mistakes

# Themen aus dem Wintersemester

- ▶ Eclipse, Grundlagen
- ▶ Statisch und Nicht-Statisch, Kommentare
- ▶ Datentypen, Casting, Operatoren
- ▶ Javadoc, Conditionals (if)
- ▶ Schleifen (for, while)
- ▶ Arrays, Strings

# Themen aus dem Wintersemester

- ▶ Eclipse, Grundlagen
- ▶ Statisch und Nicht-Statisch, Kommentare
- ▶ Datentypen, Casting, Operatoren
- ▶ Javadoc, Conditionals (if)
- ▶ Schleifen (for, while)
- ▶ Arrays, Strings
- ▶ Objekte, Felder, Methoden
- ▶ Vererbung
- ▶ Interfaces
- ▶ Datenströme
- ▶ Fehlerbehandlung



# Zum Warmwerden

## Zum Warmwerden

```

1 public class StringList {
2
3     private String[] data; // das String-Array, das intern benutzt wird
4     private int nextInsertPosition; // die nächste freie, beschreibbare Stelle des Arrays
5     public StringList(int initialSize) {
6         this.data = new String[initialSize];
7         this.nextInsertPosition = 0;
8     }
9     public void add(String s) {
10        if (full())
11            grow(); // überprüfen, ob Platz ist; wenn nicht, vergrößern (s.u.)
12        data[nextInsertPosition] = s; // den übergebenen String an die nächste freie Stelle s
13        nextInsertPosition++; // den Wert für die nächste freie Stelle um 1 erhöhen
14    }
15        if (s.equals(data[i])) {
16            return i;
17        }
18    }
19    return -1;
20 }
21 /*
22  * Eine Methode zur internen Verwendung (daher private), die das interne Array
23  * "vergrößert", bzw. (weil das ja nicht geht) ein neues, größeres Array

```

Nicht vollständig!

## Plan für's Sommersemester

- ▶ 06.04.: Einführung, Vorstellung
- ▶ 13.04.: Versionskontrolle: git, GitHub
- ▶ 20.04.: Git: Merging
- ▶ 27.04.: Iterable und Iterator
- ▶ 04.05.: Generics
- ▶ 11.05.: Java Collections: Lists, Sets
- ▶ 18.05.: Java Collections: Maps
- ▶ 25.05.: Abhängigkeitsmanagement mit Apache Maven
- ▶ 01.06.: Rekursion
- ▶ 15.06.: Effizientes Programmieren
- ▶ 29.06.: Unit-Testing mit junit
- ▶ 06.07.: Puffer
- ▶ 13.07.: Einführung in die Modulprüfung

Siehe auch: <https://uni.koeln/6LDDK>

# Exercise



`https://github.com/idh-cologne-java-2-summer-2023/exercise-01`