



Session 1: Welcome & Introduction

Fortgeschrittene Programmierung (Java 2)

Nils Reiter

`nils.reiter@uni-koeln.de`

April 10, 2024

Section 1

Introduction

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~~): Judith Nester
 - ▶ Anmeldung in Klips
- ▶ Studienleistung: Übungen via GitHub

16:00 - 17:30

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): Judith Nester
 - ▶ Anmeldung in Klips
- ▶ Studienleistung: Übungen via GitHub
- ▶ Modulprüfung
 - ▶ Programmierprojekt

Kurswebseite:

<https://uni.koeln/X4MME>



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

Plan für's Sommersemester

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

- ▶ 10.04.: Einführung, Recap, Organisatorisches
- ▶ 17.04.: Versionskontrolle mit git und GitHub
- ▶ 24.04.: Merging mit Git
- ▶ 26.04.: Iterable und Iterator
- ▶ 01.05.: entfällt: Tag der Arbeit!
- ▶ 08.05.: Generics and Lists
- ▶ 15.05.: Java Collections Framework: Queues und Sets
- ▶ 29.05.: Java Collections Framework: Maps
- ▶ 05.06.: Rekursion
- ▶ 12.06.: Abhängigkeitsmanagement mit Apache Maven
- ▶ 19.06.: entfällt
- ▶ 26.06.: Unit-Testing mit junit
- ▶ 03.07.: Sortieralgorithmen und Effizienz
- ▶ 10.07.: Allgemeine Fragerunde
- ▶ 17.07.: Einführung in die Modulprüfung

Zum Warmwerden

```
String[] wl = new String[] { "dog", "horse", ... }
```

```
class StringList {
    add(String s) { ... }
    set(int i, String s) { ... }
    delete(Strings)
    delete(int i)
}
```

Zum Warmwerden

```
1 public class StringList {
2
3     private String[] data; // das String-Array, das intern benutzt wird
4     private int nextInsertPosition;
5
6     public StringList(int initialSize) { };
7
8     public void add(String s) { };
9
10    public void set(int index, String s) { };
11
12    public void remove(String toRemove) { };
13
14    public void remove(int index) { };
15
16    public int size() { };
17 }
```

String[] = new String[3];

int nextInsertPosition = 0;

add("cat");

1.)
2.) nextInsertPosition = nextInsertPosition + 1

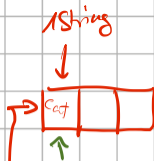
add("...")

add("...")

nIP = 3

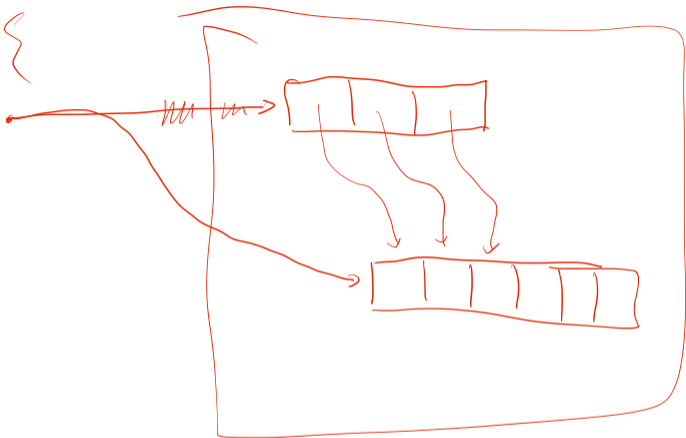
add("...")

remove(2)

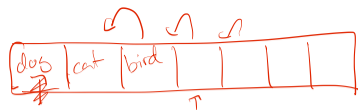


class StringList {

String[] data



remove(1)



Länge = 7

nextIP = 3



nextIP = 2

Exercise



`https://github.com/idh-cologne-java-2-summer-2024/exercise-01`