



Praktische Informatik II

Dipl.-Ing. Florian Angerer
Dipl.-Ing. Peter Hofer

florian.angerer@jku.at
peter.hofer@jku.at

Institut für Systemsoftware
Science Park 3, 2. Stock

Übungsmodus



<http://www.ssw.uni-linz.ac.at/Teaching/Lectures/PI2/2015/>

9 aus 11 Übungen mit je 24 Punkten abzugeben

- gilt als abgegeben ab 6 Punkten
- es zählen die besten 9 Übungen
- positiv ab $9 * 12 = 108$ Punkten

Abgabe bis Dienstag 8:30

- grundsätzlich Upload auf Website (siehe oben)
- 1-2 Übungen auf Papier in unserem Postkasten
(Science Park 3, 2. Stock)

Übungsmodus



Übungstests

- 2 Termine (Semesterende, nach den Sommerferien)
- 4-5 Aufgaben mit insgesamt 100 Punkten
- Antritt nur mit positiver Bewertung der Übungen
- Note ausgestellt ab drei abgegebenen Übungen

Übungsmodus

Gesamtnote

- Übungen und Test müssen positiv sein (je ab 50%)
- 50% Übungen, 50% Test

Übungsstunden

- Besprechung und tlw. Ausarbeitung der Übungen
- Keine Anwesenheitspflicht

Tutorsprechstunde

Für Fragen zur aktuellen Übung
...nicht aber zur Beurteilung

Donnerstag um 15:30 - 16:30 im
S3-209-1 "Glasbesprechungsraum"
(nahe Abgabepostkästen)

Nur nach Voranmeldung beim Tutor: [K.
Stevenson@gmx.at](mailto:K.Stevenson@gmx.at)

Übungsangabe als Deckblatt!

Übung Praktische Informatik 2

SS 2013

Übung 1: Verkettete Liste

Abgabetermin: 19.03.2013

Name: Stephen Biffmeister Matrikelnummer: 0216201

Gruppe: G1 Di 10:15 G2 Di 11:00 G3 Di 12:45

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	24	✘	Java-Programm Testfälle und Ergebnisse	Java-Programm	✘	24

Aufgabe 1: Ringliste (24 Punkte)

Implementieren Sie eine Ringliste für Objekte in der Klasse *RingList*. Die Schnittstelle ist durch die abstrakte Klasse *List* gegeben (für Methodenbeschreibungen siehe Java-Dokumentation in der Vorgabedatei).

```
package at.jku.ssw;
public abstract class List {
    public abstract void add(
        int index, Object value);
    public abstract void add(Object value);
    public abstract Object get(int index);
```

```
public abstract class Iterator {
    public abstract boolean hasNext();
    public abstract Object next();
}
```

Entwicklungsumgebung

NetBeans IDE: <https://netbeans.org/>

... oder beliebig:

- Eclipse
- IntelliJ
- Texteditor
- ...

GraphViz: <http://www.graphviz.org>

Für grafische Darstellung der Datenstrukturen

- .dot-Dateien, die in den Übungen erzeugt werden

```
digraph LinkedList {  
    rankdir=LR;  
    node001[label="a"];  
    node002[label="b"];  
    node003[label="c"];  
    node004[label="d"];  
    node001->node002  
    node002->node001  
    node002->node003  
    node003->node002  
    node003->node004  
    node004->node003  
}
```

- Verwendung: `dot -Tpng -o out.png in.dot`

