

JKU

**JOHANNES KEPLER
UNIVERSITY LINZ**

Praktische Informatik 2



Übung, 2018SS
Institut für System Software (SSW)
DI Eisl & DI Leopoldseder



Übungsmodus (1)

- Übungsaufgaben
 - **11** Übungszettel zu je **24** Punkte
 - **9 positiv** beurteilte Abgaben nötig
 - ABER: Es werden **11 Übungen gewertet**
 - ACHTUNG: Übung gilt als **abgegeben** wenn **mindestens 6 Punkte** erreicht worden sind
 - Am Ende des Semesters müssen **50%** der Punkte erreicht worden sein

$11 * 24 / 2 = 132$
- Beurteilung
 - Abschlusstest am Ende des Semesters: 03.07.2018

Übungsmodus (2)

- Beurteilung
 - Abschlusstest am Ende des Semesters (24 Punkte)
 - Endnote = $p_{\text{Test}} * 0.5 + p_{\text{Übung}} * 0.5$
 - **Beide Teile müssen positiv sein**
 - Ab der **2. abgegebenen** Übung wird eine Note ausgestellt

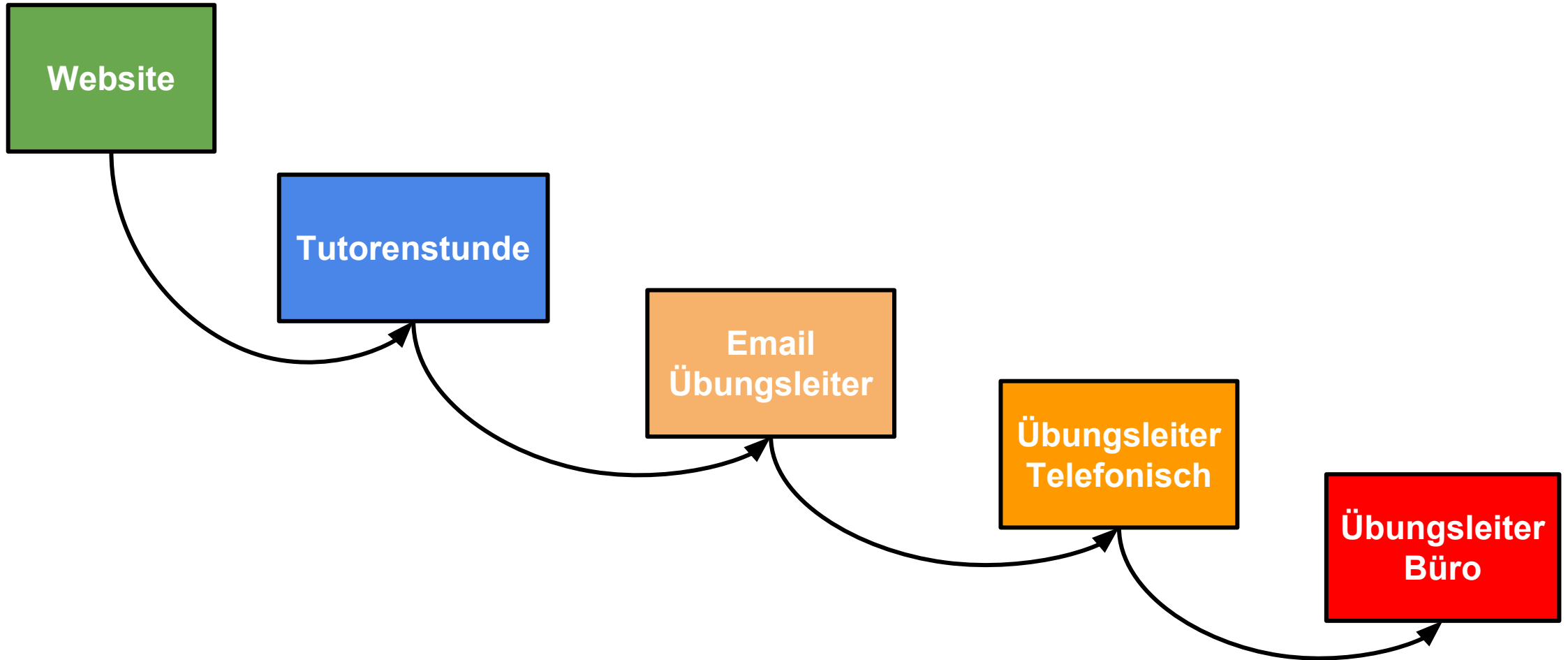
Übungsmodus (3)

- Übungsabgabe
 - Detaillierte Information auf der Kurs Website
 - Schriftliche **UND** elektronische Abgabe
 - Schriftlich: Postkasten SP3, 2. Stock
 - Elektronisch: Link auf der LVA Website
 - **Abgabe Deadline: Dienstag 08:30 laut LVA Website**
- Es herrscht **Anwesenheitspflicht**

Tutorenstunde

- Wird noch bekannt gegeben
- Fragen zu den aktuellen Übungen, **kein Pairprogramming**

Elevation of Escalation



Übungsablauf

- Alte Übungen durch besprechen
- Fragen aus der VO
- Vorbesprechung neue Übung (Aufgabenstellung)
- Vorarbeit der nächsten Übung
- Fragen ;-)

Fragen zum Übungsmodus ?



Übung 1 - Doppelt Verkettete List

Übung Praktische Informatik 2

SS 2018

Übung 1: Doppelt Verkettete Liste

Abgabetermin: 13.03.2018

Name:

Matrikelnummer:

Gruppe: G1 Di 10:15-11:00

G2 Di 11:00-11:45

G3 Di 10:15-11:00

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	24	<input type="checkbox"/>	Java-Programm, Testfälle und Ergebnisse	Projekt Archiv	<input type="checkbox"/>	

Aufgabe 1: Doppelt verkettete Liste (24 Punkte)

Implementieren Sie eine doppelt verkettete Liste für Objekte in der Klasse `LinkedList` sowie einen List Iterator in der Klasse `LinkedListIterator`. Die Schnittstellen sind durch die abstrakten Klassen `List` und `Iterator` gegeben (für Methodenbeschreibungen siehe Java-Dokumentation in dem [Vorgabeprojekt](#)).

```
package at.jku.ssw;

public abstract class List {
    public abstract void insert(int idx, Object val);
    public abstract void insertFirst(Object value);
    public abstract Object get(int index);
    public abstract int indexOf(Object value);
    public abstract int lastIndexOf(Object value);
    public abstract Object remove(int index);
    public abstract void removeFirst();
    public abstract int size();
    public abstract Iterator iterator();
}

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

Implementieren Sie die mit **TODO** markierten Funktionen in den Skeleton-Klassen `LinkedList` und `LinkedListIterator`, im Paket `at.jku.students`.

```
package at.jku.students;

public class LinkedList extends List {
    ListHead head;
    ListNode tail;
    public String makeDot() {
        return DotMaker.makeDotForLinkedList(head);
    }
    ...
}

public class LinkedListIterator extends Iterator {
    ...
}

List l = new LinkedList();
l.insertFirst('b');
l.insert(1, 'c');
l.insertFirst('f');
l.insert(1, 'e');
l.remove(0);
l.insert(0, 'a');
l.removeFirst();
l.insertFirst('d');
l.insert(2, 'o');
out.print(l.size() + ":\n");
Iterator it = l.iterator();
while (it.hasNext()) {
    out.print(" " + it.next());
}
// Output:5: d e o b c
out.open("test.dot");
out.print(((LinkedList) l).makeDot());
out.close();
```

Implementierungshinweise:

- Verwenden Sie das Vorgabeprojekt `PI2_UE01.zip`.
- Fügen Sie Ihre Implementierung in den mit **TODO** markierten Teilen der Klassen `LinkedList` sowie `LinkedListIterator` ein.
- Verwenden Sie für die Implementierung der Listenknoten die vorgegebene Klasse `ListNode` (Teil von `PI2_UE01.jar`).
- Wählen Sie für alle Skeleton Klassen, Methoden und Felder die geeignete Sichtbarkeiten (`private`, `protected`, `package`, `public`).
- Ändern Sie **keine public** Interfaces vorgegebener Skeleton Klassen (mit Ausnahme der Sichtbarkeiten).
- Halten Sie sich an die [Codierungsrichtlinien](#) auf der Kurs Website.
- Methoden mit Index als Parameter liefern eine `IndexOutOfBoundsException` bei ungültigem Index.
- Verwenden Sie die Methode `DotMaker.makeDotForList` (Teil von `PI2_UE01.jar`), um GraphViz-Bilder Ihrer Liste zu erstellen.

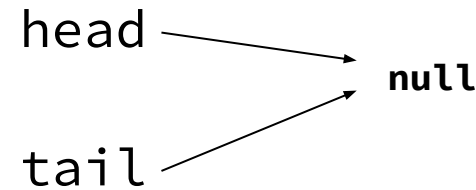
Abzugeben ist: Java-Programm, Testfälle und Ergebnisse

- Abgabetermin: 13.03.2018 08:30
- Vorgabe Projekt
- ANT zum Builden
- **TODO** markierte Methoden implementieren

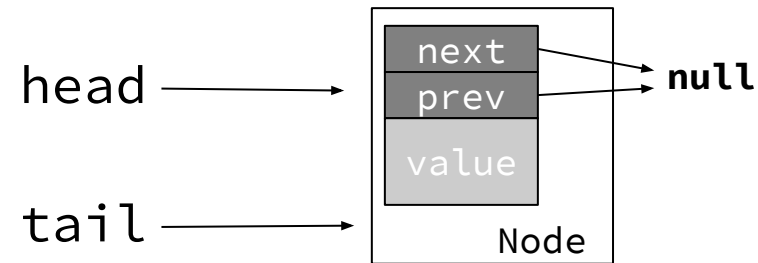
Übung 1 - Doppelt Verkettete List

- Doppelt Verkettete Liste mit *head* und *tail*
- Standard Operationen Implementieren
 - *insertFirst(value)*
 - *insert(index, value)*
 - *remove(index)*
 - ...

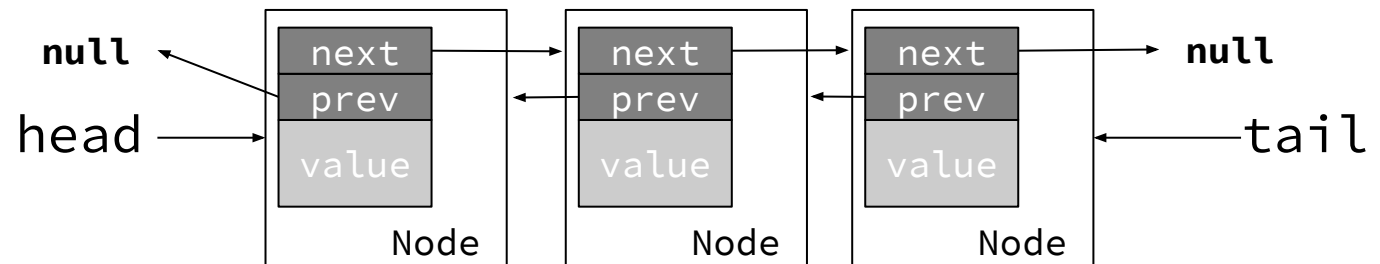
Leere Liste



Liste 1 Element



Liste *n* Element



Danke

