

# Übung 1: Doppelt Verkettete Liste

Abgabetermin: 14.03.2017

Name:

Matrikelnummer:

Gruppe:  G1 Di 10:15-11:00 G2 Di 11:00-11:45 G3 Di 12:45-13:30

| 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 insertLast(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 removeLast();
    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 {
    ListNode head;
    ListNode tail;
    public String makeDot() {
        return DotMaker.makeDotForLinkedList(head);
    }
    ...
}
public class LinkedListIterator extends Iterator {
    ...
}
```

```
List l = new LinkedList();
l.insertLast("b");
l.insert(1, "c");
l.remove(0);
l.insert(0, "a");
l.removeLast();
l.insertLast("d");
l.insert(2, "o");
Out.print(l.size() + " ");
Iterator it = l.iterator();
while (it.hasNext()) {
    Out.print(" " + it.next());
}
// Output 3: a d o
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