

Übung 1: Verkettete Liste

Abgabetermin: 10.03.2015

Name: _____ Matrikelnummer: _____

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	<input type="checkbox"/>		Java-Programm, Testfälle und Ergebnisse	<input type="checkbox"/>	

Aufgabe 1: Doppelt verkettete Liste (24 Punkte)

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

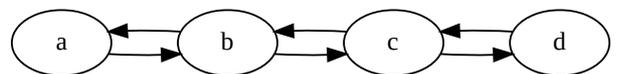
```
package at.jku.ssw;
public abstract class List {
    public abstract void insert(int idx, Object val);
    public abstract void insertLast(Object value);
    public abstract Object get(int index);
    public abstract int indexOf(Object value);
    public abstract int lastIndexOf(Object value);
    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 Klassen *LinkedList* und *LinkedListIterator* im Paket *at.jku.students*. Verwenden Sie dazu die Klassen *List*, *Iterator* und *ListNode* aus der Vorgabedatei.

```
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.insert(0, "a");
l.insertLast("d");
Out.print(l.size() + ":");
Iterator it = l.iterator();
while (it.hasNext()) {
    Out.print(" " + it.next());
}
// Ausgabe 4: a b c d
Out.open("test.dot");
Out.print(((LinkedList) l).makeDot());
Out.close();
```



test.dot

Implementierungshinweise:

- Verwenden Sie die Vorgabedatei *ssw-pi2.jar* und die Java-Dokumentation von der LVA-Website.
- Verwenden Sie für die Listenknoten die vorgegebene Klasse *ListNode*.
- Wählen Sie für alle Klassen, Methoden und Felder die geeignete Sichtbarkeiten (*private*, *protected*, *package*, *public*).
- Methoden mit Index als Parameter liefern eine *IndexOutOfBoundsException* bei ungültigem Index.
- Verwenden Sie die Methode *DotMaker.makeDotForList*, um GraphViz-Bilder Ihrer Liste zu erstellen.

Abzugeben ist: Java-Programm, Testfälle und Ergebnisse