

## Übungsblatt 5 zur Vorlesung Programmieren

Ausgabe: So 30.04.2023 – Abgabe: So 14.05.2023

### Aufgabe 1 (Abstrakter Datentyp Liste):

Implementieren Sie den Datentyp “Abstrakte Liste” für Listen ganzer Zahlen (`int`). Gehen Sie dazu wie folgt vor:

- a) Ergänzen Sie in der Datei `IntList.java` Rumpfe für die Methoden
  - `public void addFirst(int x)`
  - `public void addLast(int x)`
  - `public void remove(int x)`
  - `public boolean contains(int x)`
  - `public int size()`
  - `public boolean isEmpty()`
- b) Fügen Sie der Klasse `IntList` eine Methode `public String toString()` hinzu. Die String-Repräsentation einer Liste soll immer mit einer öffnenden eckigen Klammer beginnen, danach die Elemente durch Komma und Leerzeichen getrennt enthalten und mit einer schließenden eckigen Klammer enden. Die Liste, welche die Elemente 1, 2 und 3 (in dieser Reihenfolge) enthält, sollte daher die Repräsentation `[1, 2, 3]` besitzen, die leere Liste die Repräsentation `[]`.
- c) Fügen Sie der Klasse `IntList` einen Iterator, wie in der Vorlesung beschrieben, hinzu, der es erlaubt vorwärts durch die Liste zu gehen.
- d) **(optional)** Fügen Sie der Klasse `IntList` eine Methode `public void reverse()` hinzu. Diese soll eine Liste umkehren (d.h. das erste Element wird zum letzten etc.). Ihre Implementierung soll keinen neuen Speicherplatz mittels `new` anfordern, d.h. es soll sich um einen *in-place*-Algorithmus handeln.

Ihre Implementierung soll so geschrieben sein, dass sich die `main`-Methode der Klasse `IntListApp` ausführen lässt und das gewünschte Ergebnis liefert.

### Aufgabe 2 (Sichtbarkeit von Variablen, Klassen und Methoden):

In Ihrer Implementierung für Aufgabe 1 sind verschiedene Elemente mit den Sichtbarkeitsmodifikatoren `public` oder `private` markiert. Begründen Sie, warum Sie diese so für sinnvoll erachten oder passen Sie die Modifikatoren gegebenenfalls an.

**Aufgabe 3 (Doppelt verkettete Liste):** Modifizieren Sie die Klasse `IntList` aus Aufgabe 1 so zu einer Klasse `DoublyLinkedList`, dass doppelt verkettete Listen für die Darstellung verwendet werden. Die Methoden sollen die gleichen sein wie unter 1 a) bis c) angegeben.

Testen Sie Ihre Implementierung mit einer modifizierten Version der Klasse `IntListApp`.