

# Übungsserie Nr. 5

Ausgabe: 28. März 2012

Abgabe: 4. April 2012

## Hinweise

Für diese Serie benötigen Sie das Archiv

<http://vs.inf.ethz.ch/edu/FS2012/I2/downloads/u5.zip>.

Die Klasse *list.List* implementiert eine verkettete Liste von *int*-Werten. Sie wird in allen Aufgaben dieser Serie verwendet.

## 1. Aufgabe: (7 Punkte) Einfach verkettete Listen

Die Klasse *u5a1.Lists* bietet eine Reihe von Methoden an, um auf Listen zu arbeiten. Ihre Aufgabe ist es, diese Methoden an Hand der vorhandenen Dokumentation zu implementieren.

Ihre Implementierung muss rekursiv sein. Das heisst insbesondere, dass Sie keine *while*- oder *for*-Schleifen verwenden dürfen.

Hinweis: Die Methode *u5a1.Lists.toString* ist bereits implementiert und dient als Beispiel für eine rekursive Methode für Listen.

- *add* 1 Punkt
- *size* 1 Punkt
- *sum* 1 Punkt
- *last* 1 Punkt
- *sublist* 1 Punkt
- *valueAt* 1 Punkt
- *index* 1 Punkt

## 2. Aufgabe: (8 Punkte) Modifizierung von Listen

Die Klasse *u5a2.MutableLists* bietet eine Reihe von Methoden an, um Listen zu verändern. Ihre Aufgabe ist es, diese Methoden an Hand der vorhandenen Dokumentation rekursiv zu implementieren.

Hinweis: Verwenden Sie, falls nötig, die Methoden aus *u5a1.Lists*.

- *append* 1 Punkt
- *concat* 1 Punkt
- *insertAt* je 2 Punkte
- *remove* 2 Punkte

## 3. Aufgabe: (5 Punkte) Sortieren von Listen

Die Klasse *u5a3.SortedLists* bietet die Möglichkeit zum Sortieren von Listen. Ihre Aufgabe ist es, die Methoden an Hand der vorhandenen Dokumentation rekursiv zu implementieren.

(3a) (3 Punkte) Implementieren Sie die Methode *insertSorted*, die einen Wert in eine bereits sortierte Liste so einfügt, dass die resultierende Liste auch sortiert ist.

(3b) (2 Punkte) Implementieren Sie die Methode *sort*, die unter Verwendung von *insertSorted* die sortierte Variante einer Liste zurückliefert.

## 4. Aufgabe: (6 Punkte) Noch ein wachsender Stack

Ein Nachteil des wachsenden Stacks aus Serie 4 war, dass die Werte des Stacks immer wieder kopiert werden müssen. Eine Alternative dazu ist es, einen wachsenden Stack mit Hilfe von einfach verketteten Listen zu realisieren. So ein Stack kann dynamisch wachsen, ohne jemals kopieren zu müssen.

Die Klasse *u5a4.Stack* soll diese Idee umsetzen. Ihre Aufgabe ist es, ihre Methoden an Hand der vorhandenen Dokumentation zu implementieren.

- *push* 1 Punkt
- *pop* 2 Punkte
- *peek* 1 Punkt
- *empty* 1 Punkt
- *size* 1 Punkt