

# 1. Übung zur Vorlesung „Fortgeschrittene Programmierung“

## Simple Generics

Abgabe am Montag, 18. April – 18:00

---

Wenn Sie möchten, lesen Sie vorlesungsbegleitend diese Einführung in Java Generics:

<http://java.sun.com/docs/books/tutorial/extra/generics>

Besonders empfehlenswert sind die Abschnitte: *Generics and Subtyping*, *Wildcards*, *Generic Methods* und *More Fun with Wildcards*.

### Aufgabe 1 - Maybe

4 Punkte

Fügen Sie der `Maybe` Klasse aus der Vorlesung zwei Methoden hinzu:

1. Die Methode `toList` soll eine `java.util.LinkedList<T>` zurückliefern, die leer ist oder das gespeicherte Element enthält, wenn das Objekt, auf dem sie aufgerufen wird, vom Typ `Maybe<T>` ist.
2. Die Methode `addToList` soll eine verkettete Liste als Parameter nehmen und dieser den gespeicherten Wert hinzufügen (falls es einen gibt). Der Typ dieser Methode soll allgemein genug für die folgende Benutzung sein:

```
Maybe<Integer> opt = new Maybe<Integer>(42);  
LinkedList<Number> list = new LinkedList<Number>();  
opt.addToList(list);
```

Unter <http://www-ps.informatik.uni-kiel.de/~bjp/lehre/fopro11/Maybe.zip> kann die Implementierung von `Maybe` heruntergeladen werden.

### Aufgabe 2 - LinkedList

6 Punkte

Implementieren Sie verkettete Listen mit generischem Elementtyp in Java (ohne vordefinierte Klassen aus dem `Collections`-Framework zu verwenden). Objekte Ihrer Klasse sollen die folgenden Aufrufe erlauben:

```
LinkedList<Integer> list = new LinkedList<Integer>();  
list.add(7).add(8).add(9);  
System.out.print(list.removeNth(0));  
while (!list.isEmpty()) {  
    System.out.print(list.getFirst());  
    list = list.getRest(); }  
}
```

Neue Elemente werden vorne angefügt, wodurch sich folgende Ausgabe dieses Programms ergibt:

```
true87
```