



## Übungsblatt 11

Zu bearbeiten bis 16.02.2006

### Aufgabe 1

In der Vorlesung wurde das Paket List vorgestellt. Die Dateien des Paketes finden Sie unter

<http://www.uni-koblenz.de/~beckert/Lehre/InformatikIM/Programme/List/>

Ergänzen Sie die Klasse TList um eine Methode hat(), die die Liste so verändert, dass der Kopf vom Anfang der Liste an das Ende umgehängt wird. (hat = head after tail). Testen Sie Ihre Methode an einem geeignet Beispiel.

### Aufgabe 2

Der abstrakte Datentyp der Schlange (engl. queue) funktioniert gentlemanlike wie eine englische Warteschlange: mit der Methode put(TData data) wird ein Element an das Ende der Schlange eingereiht, mit TData get() wird das erste Element aus der Schlange entnommen. Das Prinzip ist unter dem Kürzel „fiffo“ bekannt, say first in, first out, auf Deutsch: wer zuerst kommt, mahlt zuerst.

Ergänzen Sie das Paket List um eine Klasse TQueue, die diesen Datentyp implementiert.

```
public class TQueue {  
  
    public TQueue() ... // der Konstruktor, der die leere Liste erzeugt.  
    public boolean isEmpty() ... // liefert genau dann true, wenn die Schlange leer ist.  
    public void put(TData data) ... // fügt ein Element mit Inhalt dat an das Ende  
    public TData get() ... // liefert den Kopf der Schlange und entfernt das Element  
  
}
```

Schreiben Sie ein geeignetes Testprogramm für die Klasse TQueue.

### Aufgabe 3

Einen Stack kann man nutzen, um eine Liste umzukehren. Man legt die Listenelemente auf den Stapel, und nimmt sie dann in umgekehrter Reihenfolge wieder herunter und fügt sie erneut zu einer Liste zusammen. Ergänzen Sie die Klasse ZList um eine Methode TList stackreverse(TList list), die eine umgedrehte List nach dieser Methode erzeugt.

Schreiben Sie ein Testprogramm, das die Liste (4,7,1,1) erzeugt und mit stackreverse umkehrt und ausgibt.