Institut für Informatik
der Universität München
Prof. Dr. M. Hofmann
Dr. J. Johannsen, Dr. R. Matthes

SS 2004

24.6.04

Übungen zur Vorlesung
Informatik II

Blatt 9

Abgabe der Hausaufgaben bis spätestens am 5.7.04, 14:00 Uhr über
`http://lehre.tcs.ifi.lmu.de/info2/Abgabe/abgabe.php`,
Bearbeitung in Gruppen zu max. 3 Personen ist zulässig.

Bitte halten Sie sich an folgende Grundsätze zur Bearbeitung der Aufgaben. Die Bewertung Ihrer Abgaben kann andernfalls deutlich herabgesetzt werden.

Verwenden Sie genau die Signatur der Aufgabenstellung, insbesondere die Namen der Klassen und Methoden in der vorgegebenen Klein- und Großschreibung.

Verwenden Sie keine „magischen" Größen in Ihren Programmen, d. h., weisen Sie Konstanten (außer den Zahlen 0 und 1 und dem leeren String) immer einer mit `final` deklarierten und in Großbuchstaben geschriebenen „Variablen" zu und verwenden Sie anschließend nur diese. Wer gute Gründe hat für eine Abweichung von dieser Regel, der kommentiere seinen Code entsprechend.

Jede Klasse und `public`-Methode muß angemessen mit `javadoc`-Kommentaren dokumentiert werden.

Programmieraufgabe P-21 (`Generator.java, GeneratorTest.java`): **5 Punkte**
Schreiben Sie eine Unterklasse `Generator` der Klasse `java.util.Random` mit einer zusätzlichen Methode

```
public static Generator generator()
```

Implementieren Sie diese unter Verwendung des Singleton-Patterns in einer solchen Weise, daß beim Aufruf von `generator` immer derselbe Zufallszahlengenerator zurückgegeben wird und nicht jedesmal ein neuer.

Auf der Vorlesungs-Homepage finden Sie eine Klasse `GeneratorDumm`, die zufällige Folgen von Nullen und Einsen produzieren will, dies aber nicht leistet, wie Sie leicht erkennen können. Schreiben Sie eine Klasse `GeneratorTest`, die dies richtig tut unter Verwendung von `Generator`. Dabei soll insbesondere `testedumm` durch eine ebenfalls *rekursive* Methode

```
public static void teste(int n)
```

ersetzt werden, die dann das tut, was man von `testedumm` eigentlich erwartet hätte.

– bitte wenden –

Programmieraufgabe P-22 (`Umdrehen.java`): **5 Punkte**
Schreiben Sie eine Klasse `Umdrehen` mit einer Methode

```
public static void reverse(InfoIIListe l)
```

Das Interface `InfoIIListe` dazu finden Sie auf der Vorlesungs-Homepage. Dieses Interface ist ein Teil dessen, was von der Klasse `java.util.LinkedList` implementiert wird. `reverse` soll die gegebene Liste `l` umdrehen, wenn alle Methoden des Interfaces sich so verhalten, wie das im Javadoc zu `java.util.LinkedList` beschrieben ist. Dazu werden Sie ebenfalls die Dokumentation zu `java.util.ListIterator` konsultieren müssen.

Hinweis: Sie können Ihre Methode mit `UmdrehenTest.java` von der Vorlesungs-Homepage testen.

Programmieraufgabe P-23 (`LinkedList.java`): **10 Punkte**

Sie finden auf der Vorlesungs-Homepage eine teilweise unvollständige, wenn auch kompilierende Implementierung der Klasse `LinkedList` und ihrer Hilfsklasse `LinkedListIterator`.

Vervollständigen Sie diese, in dem Sie alle mit `keine Implementierung` gekennzeichneten Methoden durch eine sinnvoll funktionierende Implementierung ersetzen.

Das zum Interface `InfoIIListe` und zu den Bibliotheksklassen `java.util.LinkedList` und `java.util.ListIterator` Gesagte gilt dabei sinngemäß wie in der vorhergehenden Aufgabe.

Hinweis: Für die Methode `nextIndex` in `LinkedListIterator` bietet es sich an, eine zusätzliche Instanzvariable für die Cursor-Position einzuführen.

Hier noch der Inhalt von `InfoIIListe.java`:

```
import java.util.ListIterator;

public interface InfoIIListe{

    ListIterator listIterator();
    Object getFirst();
    Object getLast();
    void addFirst(Object obj);
    void addLast(Object obj);
    Object removeFirst();
    Object removeLast();
}
```