

Übungsblatt 9

Abgabe bis Freitag, 11.01.2012, 12:00 Uhr

Hinweis: Lösungen immer per Email an den zuständigen Tutor schicken. Die Emailadressen sind auf der Homepage zur Vorlesung gelistet. Verwenden Sie für alle Programmieraufgaben die Struktur des Beispielprojekts ¹. Kompilieren Sie Ihre Projekte mit Hilfe des ant-Buildsystems.

Aufgabe 9.1

Betrachten Sie den Auszug der Klasse `Tuple`, die ein Tupel aus zwei Objekten darstellt.

```
public class Tuple<A,B> {
    public void setA( ... ) { ... }
    public void setB( ... ) { ... }

    public ... getA(){ ... }
    public ... getB(){ ... }

    public String toString() {
        ...
    }

    private A a;
    private B b;
}
```

1. Vervollständigen Sie die vorgegebenen Methoden der Klasse `Tuple`.
2. Vervollständigen Sie die `toString` Methoden der Klasse `Tuple`, so dass sie einen String der Form “(···,···)” ausgibt.
Hinweis: Verwenden Sie die `toString`-Methoden der zwei Member-Variablen.
3. Erweitern Sie die Klasse `Tuple<A, B>` so, dass sie das Interface `Comparable<Tuple<A, B>>` implementiert. Hierbei soll zunächst das erste Element der Tupel mittels der Methode `compareTo` verglichen werden. Sind diese gleich, sollen die zweiten Elemente verglichen werden.
4. Testen Sie Ihre Implementierung mit der Test-Klasse `TupleTest`, die Sie auf der Vorlesungshomepage finden.

¹<http://ais/teaching/ws12/info/material/MyProject.zip>

Aufgabe 9.2

In dieser Aufgabe soll eine zyklisch verkettete Liste implementiert werden. Im Gegensatz zur Liste, die in der Vorlesung vorgestellt wurde, sind die Elemente dieser Datenstruktur wie in Abbildung 1 dargestellt organisiert.

Implementieren Sie eine Klasse `CyclicLinkedList`, die eine zyklisch verkettete Liste repräsentiert. Eine Instanzvariable `current` zeigt dabei immer auf das aktuelle Element der Liste. Sie können die Klasse `Node` aus der Vorlesung verwenden.

Implementieren Sie folgende Methoden:

- `boolean isEmpty()`
- `void insert(Object o)`
- `void removeCurrent()`
- `Node searchNode(Object o)`

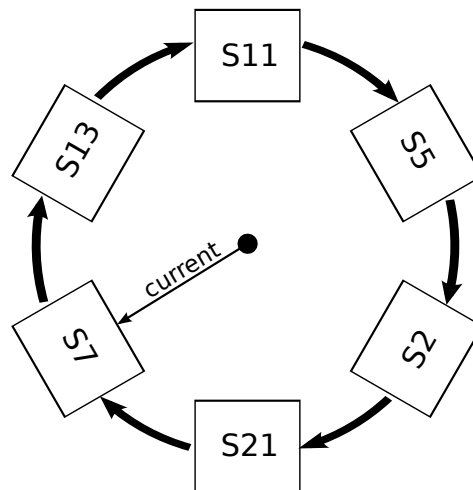


Abbildung 1: `CyclicLinkedList`