

Aufgabe 2: Das Listenmodell funktionaler Sprachen, JUnit-Test

Abgabetermin: 13. Okt. 2014

Gegeben ist die Java-Klasse `ueb02.List`.

- Implementieren Sie in mindestens einer der Sprachen Lisp, ML, Haskell, Scala oder Java eine Funktion `prepend` und eine Funktion `attach` mit jeweils den Parametern `value` und `list`. `prepend` soll vor die einfach verkettete Liste einen weiteren Knoten mit dem Wert `value` stellen.
`append` soll ans Ende der einfach verketteten Liste `list` einen weiteren Knoten mit dem Wert `value` anhängen.
Die Implementierung darf im rein funktionalen Stil keine änderbaren Variablen enthalten. In Java bedeutet dies, alle Variablen müssen **final** sein. Beide Funktionen ändern daher nicht die übergebene Liste, sondern erzeugen aus den übergebenen Parametern eine neue Liste.
- Schreiben Sie einen Unit-Testtreiber, der diese beiden Funktionen mit jeweils einer leeren Liste und einer mehrelementigen Liste überprüft.

Die Sprachen Lisp, ML, Haskell, Scala verfügen über vordefinierte Fähigkeiten für einfach verkettete Listen. Wenn Sie die Aufgabe in Java lösen wollen, benutzen Sie bitte die ausgegebene Klasse `ueb02.List`:

```
package ueb02;

/**Einfach vorwärts verkettete Liste von Ganzzahlen à la LISP.
 * Eine Liste mit den Elementen 1, 2 und 3 kann gebildet werden durch
 * new List(1, new List(2, new List(3,null))*/
public class List {

    /**Inhalt eines Listenknotens*/
    public final int head;

    /**Vorwärtsreferenz zum nächsten Listenknoten.
     * Das Listenende wird durch null markiert.*/
    public final List tail;

    /**Dieser Konstruktor entspricht der CONS-Operation von LISP.*/
    public List(final int head, final List tail){
        this.head = head;
        this.tail = tail;
    }

    /**Liefert die String-Darstellung einer aus Paaren gebildeten,
     * über die Referenz tail vorwärts verketteten Liste.*/
    public String toString(){
        return Integer.toString(head) + (tail==null ? "" : " " + tail.toString());
    }
}
```

Die Signaturen der Methoden lauten in Java:

```
/**Prepends the given value to the front of the given list.*/
public List prepend(final int value, final List list)

/**Attaches the given value to the end of the given list.*/
public List attach(final int value, final List list)
```