

## Übungen zu Einführung in die Informatik I

### Aufgabe 48      **Doppelt verkettete Liste in Java**

Doppelt verkettete Listen haben gegenüber normalen Listen den Vorteil, dass jedes Element zusätzlich zum Nachfolger auch seinen Vorgänger kennt.

- a) Erstellen sie in UML einen Entwurf für eine doppelt verkettete Liste. Diese soll folgende Anforderungen erfüllen:
  - Die Liste soll Elemente vom Typ `Object` aufnehmen können.
  - Die Liste soll folgende Operationen effizient (konstanter Zugriff) unterstützen: `isEmpty`, `addFirst`, `addLast`, `getFirst`, `getLast`, `removeFirst`, `removeLast`, `getSize`
- b) Implementieren Sie das UML-Model in Java.

### Aufgabe 49      **Implementierung eines sortierbaren Bücherregals mit `java.util.Vector<E>`**

Das in den vorangegangenen Aufgaben implementierte Bücherregal soll nun mit von Java zur Verfügung gestellten Mitteln umgesetzt werden. Für die Umsetzung der nachfolgenden Aufgaben können Sie die folgende API-Dokumentation zu Hilfe nehmen:

<http://java.sun.com/j2se/1.5.0/docs/api/java/util/Vector.html>

<http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Comparable.html>

- a) Erweitern Sie die bisherige Klasse für Bücher um eine natürliche Ordnung, basierend auf dem Buchtitel. Implementieren Sie hierfür die `compareTo` Funktion aus `Comparable` geeignet.
- b) Entwickeln Sie eine Klasse für Bücherregale, welche die Verwaltung der Bücher über die Java-Klasse `Vector<E>` vornimmt und folgende Funktionen ermöglicht:
  - Ausgabe der Anzahl der derzeit enthaltenen Bücher.
  - Einfügen eines Buches in das Bücherregal.
  - Auffinden eines Buches basierend auf dem Buchtitel.
  - Entnehmen eines Buches durch Angabe des Buchtitels.
  - Ausgabe der Liste aller derzeit enthaltenen Bücher.
- c) Um das Bücherregal bei Bedarf zu sortieren soll eine Funktion `sort()` zur Verfügung gestellt werden. Verwenden Sie hierfür ebenfalls die für die Klasse `Vector<E>` bereitgestellte Funktionalität.
- d) Testen Sie Ihre Implementierung geeignet.

### Aufgabe 50      Sprachen und Grammatiken

Gegeben sei die Sprache  $L = \{a^m b^n : m, n \geq 1\}$ .

- Schreiben Sie eine Grammatik  $G = (N, T, P, S)$  auf, welche diese Sprache erzeugt.
- Zeichnen Sie den Ableitungsbaum für das Wort  $a^3 b^2$  auf.
- Implementieren Sie eine OCaml-Funktion `check`, welches ein übergebenes Wort auf seine Zugehörigkeit zu der oben angegebenen Sprache überprüft.
- Testen Sie Ihre Implementierung geeignet.

### Aufgabe 51 (H) Polnische Notation

(4+4+2 Punkte)

Der polnische Mathematiker Jan Lukasiewicz entwickelte in den 1920er-Jahren die Polnische Notation. Hierbei handelt es sich um eine klammerfreie Schreibweise für Formeln bzw. allgemein Ausdrücke, bei welcher der Operator vor seinen Operanden geschrieben wird:

$$\begin{array}{ccccccc} 21 & + & 43 & \rightarrow & + & 21 & 43 \\ (4 & + & 9) & : & (17 & + & 2) \rightarrow : & + & 4 & 9 & + & 17 & 2 \end{array}$$

- a) Gegeben sei eine Liste von Elementen folgender Typdefinition:

```
type polList = Number of int | Add | Div | Mul | Sub;;
```

- Schreiben sie nun eine rekursive Funktion `collapseList` in OCaml, welche die Auswertung eines Schrittes für einen Term der Polnischen Notation übernimmt. Suchen sie dafür die erste Stelle in der Liste an der sich zunächst ein Operator gefolgt von zwei Zahlen befindet. Ersetzen sie diese drei Elemente durch das Ergebnis der Operation. So wird z.B. `* + 21 43 2` zu `* 64 2`.
- Implementieren Sie nun eine rekursive Funktion `evalList`, die unter zu Hilfenahme der Funktion `collapseList` die Liste solange reduziert bis nur noch das Ergebnis in der Liste enthalten ist und dieses zurückliefert.

- b) Verwenden Sie die Typdefinition

```
type polOperator = Add | Div | Mul | Sub;;  
type polTerm = Result of int  
              | PolTerm of (polOperator * polTerm * polTerm);;
```

- und schreiben Sie eine Funktion `evalTerm`, welche einen übergebenen Term der Polnischen Notation rekursiv auswertet.
  - und schreiben Sie eine Funktion `termToString`, welche einen Term in Polnischer Notation von der obigen Typdefinition in einen String umwandelt.
- c) Entwerfen Sie eine Grammatik für die Polnische Notation.