— Übungsblatt 5 (P) —

Aufgabe 12 (P) (Suchen)

Laden Sie sich vom Programm-Vorlagen-Verzeichnis auf der Moodle-Plattform die Klasse `ListeL` herunter. Es handelt sich dabei um die in der Vorlesung behandelte Klasse, wobei aufgrund von

```
public class ListeL<I extends Comparable<I>>
```

der Datentyp-Platzhalter `I` für den Inhalt der Listenelemente die Vergleichbarkeit sicherstellen muss. Bei der Erzeugung von Listen-Objekten kommen daher für `I` nur Klassen in Frage, die das `Comparable`-Interface implementieren, so dass stets Vergleiche der Listenelemente mit der Methode `compareTo` möglich sind.

Ergänzen Sie die Klasse `ListeL` um eine Methode

```
public int find1 (I o)
```

die eine sequentielle Suche auf der Liste durchführt und die Position (≥ 0) zurückliefert, an der `o` zum ersten Mal in der Liste vorkommt, oder -1, falls `o` nicht in der Liste vorkommt.

Schreiben Sie auch ein Testprogramm, das die neue Methode verwendet.

Aufgabe 13 (P) (Sortieren)

Ergänzen Sie die Klasse `ListeL` um eine Methode

```
public void sort1()
```

die die Liste mittels *Selectionsort* sortiert. Dabei sollen die zu tauschenden Elemente nicht aus der verketteten Liste ausgehängt werden, sondern lediglich ihre Inhalte vertauscht werden.

Schreiben Sie auch ein Testprogramm, das die neue Methode verwendet.

Aufgabe 14 (P) (Sortieren)

Ergänzen Sie die Klasse `ListeL` um eine Methode

```
public void sort2()
```

die die Liste mittels *Bubblesort* sortiert. Dabei sollen die zu tauschenden Elemente nicht aus der verketteten Liste ausgehängt werden, sondern lediglich ihre Inhalte vertauscht werden.

Schreiben Sie auch ein Testprogramm, das die neue Methode verwendet.

Aufgabe 15 (P) (Sortieren)

Ergänzen Sie die Klasse `ListeL` um eine Methode

```
public void sort3()
```

die die Liste mittels *Insertionsort* sortiert. Dabei soll das Element das im bereits bearbeiteten Teil der Liste eingefügt werden muss, zunächst aus der verketten Liste ausgehängt und danach an der richtigen Position im bereits bearbeiteten Teil wieder eingehängt werden.

Schreiben Sie auch ein Testprogramm, das die neue Methode verwendet.

Aufgabe 16 (P) (Sortieren)

Ergänzen Sie die Klasse `ListeL` um eine Methode,

```
public void sort4() {
 return mergeSort(this); // Aufruf von Mergesort
}
```

die die Liste mittels *Mergesort* sortiert.
Schreiben Sie auch ein Testprogramm, das die neue Methode verwendet.