

Name: \_\_\_\_\_

Tutor: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Punkte: \_\_\_\_\_

Gruppe: \_\_\_\_\_

Abgabe am: Di, 25. 3. 2002 10:00

## Wortliste

Implementieren Sie eine einfach verkettete, lineare Liste, die die Häufigkeit der eingefügten Wörter zählt und statistische Auswertungen ermöglicht. Es soll nicht zwischen Groß- und Kleinschreibung unterschieden werden.

Achten Sie darauf, dass nur die angegebenen Methoden *public* sind, alles andere soll *private* deklariert sein (oder zumindest gut begründet wenn nicht).

```
public class WordList {
    public WordList() { ... }
    public void insert(String word) { ... }
    // Get frequency of word
    public int getFrequency(String word) { ... }
    // Number of different words
    public int nOfDifferentWords() { ... }
    // Mean frequency of equal words
    public double meanWordCount() { ... }
    // Get all words with specified frequency
    public String[] getWords(int frequency) { ... }
    // Get all words starting with the specified prefix
    public String[] getWordsStartingWith(String prefix) { ... }
    public void printList() { ... }
}
```

Beispiel: Einlesen von "To be or not to be" soll beim Aufruf der Methode `printList()` in etwa folgende (Test-)Ausgabe liefern (die Reihenfolge der Wörter ist nicht relevant):

```
1 NOT (1 times)
2 BE (2 times)
3 OR (1 times)
4 TO (2 times)
```

Die Methode `nOfDifferentWords()` liefert in diesem Beispiel das Ergebnis 4, `meanWordCount()` liefert 1.5.

- a) Implementieren Sie eine "einfache" Wortliste laut obiger Spezifikation und Erweitern Sie das Testprogramm, das in einem Durchlauf die Funktionalität der Wortliste überprüft (siehe Vorlage). Testen Sie Funktionalität und Effizienz Ihrer Lösung.
- b) Überlegen Sie sich 2 verschiedene Optimierungen für die Wortliste (basierend auf einer linearen Liste). Versuchen Sie dabei beiden Varianten möglichst unterschiedliche Ideen/Schwerpunkte zugrunde zu legen. Implementieren Sie diese Optimierungen (jeweils in einer eigenen Klasse) und Testen Sie die Optimierungen auf Korrektheit (siehe Punkt a) und anschließend auf Effizienz (siehe Vorlage). Überlegen Sie (schriftlich):
  - Wie schlagen die Optimierungen an?
  - Lässt sich überhaupt so etwas wie eine "Gesamteffizienz" finden? Wenn ja, wie kann man sie definieren? Wenn nein, warum nicht?
  - Gehen Optimierungen für eine Methode auf Kosten anderer Methoden? Wie lassen sich entsprechende negative Einflüsse erklären?
  - Lassen sich die obigen Optimierungen miteinander kombinieren oder schließen sie sich gegenseitig aus?