

Übung 02: Ringliste

Abgabetermin: 23.03.2010 10:15

Name: _____

Matrikelnummer: _____

Gruppe: G1 (Wolfinger) G2 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 02.1	24	<input type="checkbox"/>	Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	

Aufgabe 02.1: Stichwortverzeichnis als sortierte Ringliste mit Kopfknoten

Implementieren Sie ein Programm das für einen Text ein Stichwortverzeichnis aufbaut. Lesen Sie den Text Wort für Wort aus der Vorgabedatei *Alice.txt* und tragen Sie für jedes Wort in das Stichwortverzeichnis ein, in welchen Absätzen dieses Wort vorkommt. Absatzwechsel erkennen Sie am *LineFeed*-Trennzeichen '\n'. Kommt ein Wort in einem Absatz mehrfach vor, werden die Vorkommen mehrfach erfasst.

```

1  Alice's Adventures in Wonderland
2
3  Lewis Carroll
4
5  CHAPTER I
6  Down the Rabbit-Hole
7  Alice was beginning to get very tired of sitting by her sister on the bank,
   and of having nothing to do: once or twice she had peeped into the book her
   sister was reading, but it had no pictures or conversations in it, `and what
   is the use of a book,' thought Alice `without pictures or conversation?' So
   she was considering in her own mind (as well as she could, for the hot day
   made her feel very sleepy and stupid), whether the pleasure of making a
   daisy-chain would be worth the trouble of getting up and picking the daisies,
   when suddenly a White Rabbit with pink eyes ran close by her.
8  There was nothing so very remarkable in that; nor did Alice think it so
   very...
```

Implementieren Sie das Stichwortverzeichnis als Liste (*WordIndexImpl*) von Listen (*WordDataImpl*). Implementieren Sie alle Listen als sortierte Ringlisten mit Kopfknoten. Die Knoten (*WordNode*) von *WordIndexImpl* verweisen auf Objekte der Klasse *WordDataImpl* (siehe Abbildung 1). Die Knoten (*ParagraphNode*) von *WordDataImpl* enthalten die Absatznummern.

Implementieren Sie die Klasse *WordDataImpl* mit folgender Schnittstelle:

```

interface WordData {
    String getWord();
    int getFrequency();
    int[] getParagraphs();
}
```

- *String getWord()* liefert das Wort
- *int getFrequency()* liefert die Häufigkeit für ein Wort
- *int[] getParagraphs()* liefert ein Array mit den Nummern der Absätze in denen das Wort vorkommt

Implementieren Sie die Klasse *WordIndexImpl* mit folgender Schnittstelle:

```

interface WordIndex extends Iterable<WordData> {
    void insert(String word, int paragraph);
    int getFrequency(String word);
    int getDifferentWordCount();
    int getTotalWordCount();
    double getMeanWordFrequency();
    String[] getWordsWithMinFrequency(int frequency);
    String[] getWordsStartingWith(String prefix);
    Iterator<WordData> iterator();
}
```

- `void insert(String word, int paragraph)` fügt ein neues Wort ein
- `int getFrequency(String word)` ermittelt die Häufigkeit für ein Wort
- `int getDifferentWordCount()` liefert die Anzahl der verschiedenen Wörter
- `void getTotalWordCount()` liefert die Summe der Häufigkeiten aller Wörter
- `double getMeanWordFrequency()` liefert die durchschnittliche Häufigkeit aller Wörter
- `String[] getWordsWithMinFrequency(int frequency)` liefert alle Wörter mit einer gegebenen Mindesthäufigkeit
- `String[] getWordsStartingWith(String prefix)` liefert alle Wörter die mit einem gegebenen Prefix beginnen.
- `Iterator<WordData> iterator()` liefert einen Iterator für alle Wörter

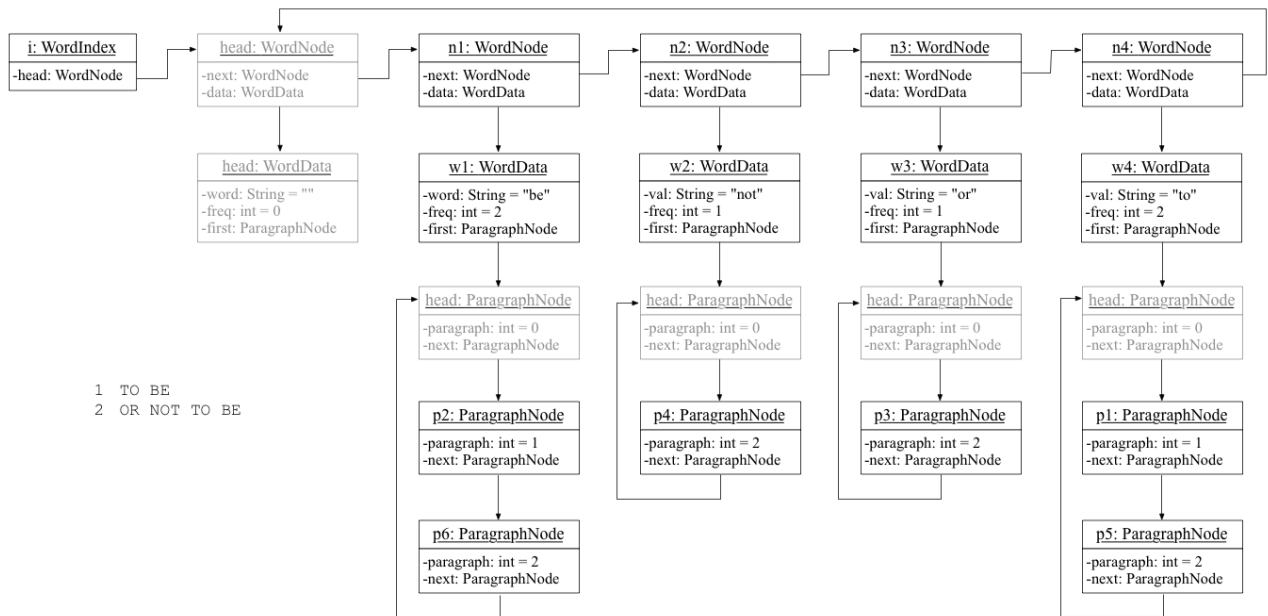


Abbildung 1) UML-Objektdiagramm für Datenstruktur *WordIndex*

Beispiel:

```

WordIndex index = new WordIndexImpl();
index.insert("TO", 1);
index.insert("BE", 1);
index.insert("OR", 2);
index.insert("NOT", 2);
index.insert("TO", 2);
index.insert("BE", 2);
for(WordData w: index) {
    Out.print(String.format("\n%s (%d times):",
        w.getWord(), w.getFrequency()));
    for(int paragraph: w.getParagraphs()) {
        Out.print(" " + paragraph);
    }
}
    
```

Ausgabe:

```

BE (2 times): 1 2
NOT (1 times): 2
OR (1 times): 2
TO (2 times): 1 2
    
```

Abzugeben ist: Java-Programm, Testergebnisse