

Übung 02: WortListe

Abgabetermin: 27.03.2007 12:00

Name: _____

Matrikelnummer: _____

Gruppe: G1 (Dhungana) G2 (Wolfinger) G3 (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: Wortliste als Ringliste mit Kopfknoten

Implementieren Sie in der Klasse `SortedWordList` eine sortierte Ringliste mit Kopfknoten, die die Häufigkeit der eingefügten Wörter zählt und statistische Auswertungen ermöglicht. Es soll nicht zwischen Groß- und Kleinschreibung unterschieden werden.

```

class SortedWordList
    implements WordList {
    private WordNode head;
    public SortedWordList () { ... }
    ...
}

class WordNode {
    int count; // frequency
    String word;
    WordNode next; // next node in list
    public WordNode (...) {...}
}

```

Die Klasse `SortedWordList` soll das folgende Interface `WordList` implementieren:

```

public interface WordList {
    // Insert a word
    void insert(String word);
    // Remove a word
    void remove(String word);
    // Get frequency of word
    int getFrequency(String word);
    // Number of different words
    int nOfDifferentWords();
    // Mean frequency of equal words
    double meanWordCount();
    // Get all words with specified frequency
    String[] getWords(int frequency);
    // Get all words starting with the specified prefix
    String[] getWordsStartingWith(String prefix);
    void print();
    // Creates a copy
    WordList clone();
    // Merges two lists
    void merge(WordList list);
    // Gets an iterator
    Iterator iterator();
}

```

Beispiel: Einlesen von "To be or not to be" soll beim Aufruf der Methode `print` in etwa folgende (Test-)Ausgabe liefern (Reihenfolge aufsteigend sortiert):

```

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

```

Die Methode `nOfDifferentWords` liefert in diesem Beispiel das Ergebnis 4, `meanWordCount` liefert 1.5 (versuchen Sie bei der Implementierung ohne Schleifen auszukommen).

Implementierungshinweis:

- Die Methode *merge* ist so zu implementieren, dass die Knoten der übergebenen Liste *list* dupliziert und in die durch *this* referenzierte Liste einsortiert werden. Verwenden Sie dazu nicht die *insert*-Methode, sondern duplizieren Sie die Knoten aus *list* und ketten Sie die duplizierten Knoten in *this* ein. Der Eingangsparameter *list* ist vom Typ *WordList* (interface), Sie haben daher keinen Zugriff auf den *head*-Knoten. Verwenden Sie stattdessen den Iterator um die übergebene Liste zu durchlaufen.
- Testen Sie Ihre Implementierung mit dem JUnit-Test *TestWordList.java* (als Download auf der Übungsseite im Internet). JUnit 4.0 bekommen Sie unter <http://www.junit.org>.

Abzugeben ist:

- Das Java-Programm
- Testfälle und die Ergebnisse