

## Übungsaufgabe Rekursion mit Listen

In dieser Übungsaufgabe soll Rekursion anhand einer Liste von Strings geübt werden. Die dafür notwendigen Knoten sind in der Klasse `WordNode` mit den üblichen Settern und Gettern bereits gegeben.

Drei Methoden sind hier bereits in einer iterativen Version vorgegeben und sollen nicht weiter verändert werden.

Die gleiche Funktionalität soll stattdessen durch drei entsprechende Paare von Methoden mit Rekursion erreicht werden.

Falls Sie mit dem Thema Rekursion noch unsicher sind, so finden Sie ganz unten in der Klasse ein ausführliches Beispiel. Hier wird nochmal ausführlich erklärt wie Rekursion funktioniert und an einer Beispielaufgabe aussehen könnte.

In der `main`-Methode können Sie Ihre Ergebnisse Testen, oder sich natürlich auch eigene Testfälle erstellen.

### Aufgabe 1 `appendWord`

Die Methode `appendWord(String word)` soll den als Parameter übergebenen String ans Ende der Liste anhängen. Finden Sie mit der Methode `appendWord(String word, WordNode node)` auf rekursive Weise das letzte Element der Liste und hängen das neue Element an das letzte Element an. Starten Sie die Rekursion durch einen geeigneten Aufruf in der Methode `appendWord(String word)`. Beachten Sie den Sonderfall, dass die Liste leer ist. Fangen Sie diesen Sonderfall in der Methode `appendWord(String word)` ab.

Beispiel:

"Diese" --> "Aufgabe" --> "ist" --> null

Aufruf von `appendWord("gruen")` --> neue Liste:

"Diese" --> "Aufgabe" --> "ist" --> "gruen" --> null

Die Ausführung soll den gleichen Effekt haben wie die Ausführung der Methode `appendWordIterative(String word)`.

## Aufgabe 2 `getWordAt`

Die Methode `getWordAt(index)` soll den String an der durch den Parameter `index` gegebenen Stelle zurückgeben. Das erste Element hat dabei wie bei Arrays den Index 0. Finden Sie mit der Methode `getWordAt(int index, WordNode node)` auf rekursive Weise die gewünschte Stelle und geben Sie den entsprechenden String zurück. Starten Sie die Rekursion durch einen geeigneten Aufruf in der Methode `getWordAt(int index)`. Existiert kein Element mit dem gewünschten Index, soll die Rückgabe null sein.

Beispiel:

"Ich"(0) --> "habe"(1) --> "droelf"(2) --> "Elemente"(3)

Rückgabe von `getWordAt(2)` :

"droelf"

Die Rückgabe soll identisch mit der Rückgabe der Methode

`getWordAtIterative(int index)` sein.

## Aufgabe 3 `toString`

Die Methode `toString()` soll alle Worte der Liste durch je ein Leerzeichen getrennt in einem String zurückgeben. Bauen sie mit der Methode `toString(WordNode node)` den String rekursiv zusammen und starten Sie die Rekursion durch einen geeigneten Aufruf in der Methode `toString()`. Falls die Liste leer ist, soll ein leerer String (also "") zurückgegeben werden.

Beispiel:

"Das" --> "ist" --> "eine" --> "Liste" --> null

Rückgabe von `toString`:

"Das ist eine Liste"

Die Rückgabe soll identisch mit der Rückgabe der Methode `toStringIterative()` sein.