Aufgabe 1 (revidiert)

Ausgabedatum: 15.03.2007
Abgabetermin: 22.03.2007 (ursprüngliche Aufgabe)
Abgabetermin: 29.03.2007 (revidierte Aufgabe)

Transformation eines Textes, sodass die Buchstaben jedes 2. Wortes in umgekehrter Reihenfolge erscheinen.

Die in der Vorlesung besprochene Aufgabenstellung der Texttransformation soll in drei Varianten in Java implementiert werden:

1. Implementierung des in der Vorlesung entwickelten (nicht-rekursiven) Algorithmus.
2. Implementierung eines eigenen OO-Designs (z.B. eine eigene Klasse für ein Wort, etc.)
3. Implementierung des in der Vorlesung vorgestellten Programms, das auf schrittweiser Verfeinerung und einem rekursiven Algorithmus beruht.

Zur Erinnerung nochmals die Aufgabenstellung (in Anlehnung an [EWD302] bzw. [Dijkstra89]):

Gegeben sei ein Zeichensatz bestehend aus:

- Buchstaben
- Leerzeichen
- Punkt

Wörter bestehen aus max. 20 Buchstaben und werden durch beliebig viele Leerzeichen getrennt. Die Eingabe besteht aus beliebig vielen Wörtern und Lehrzeichen und wird durch einen Punkt abgeschlossen.

Die Ausgabe soll wie folgt aussehen:

1. Lehrzeichen und der abschließende Punkt sollen 1:1 wie bei der Eingabe ausgegeben werden.
2. Die Buchstaben jedes 2. Wortes sollen in umgekehrter Reihenfolge sein.
   Genauer: Gegeben sei eine Ordnung der Wörter in Leserichtung (d.h. von links nach rechts), sodass jedem Wort eine eindeutige Zahl 1,2,3... zugeordnet werden kann. Wörter denen eine ungerade Zahl zugeordnet ist sind exakt wiederzugeben. Bei Wörtern denen eine gerade Zahl zugeordnet ist sind die Buchstaben in umgekehrter Reihenfolge wiederzugeben.

Beispiel (revidiert): "␣␣␣this␣␣␣is␣␣a␣silly␣␣program␣␣." wird transformiert in "␣␣␣this␣␣␣si␣␣a␣yllis␣␣program␣␣."

Referenzen:

[EWD302] Dijkstra, E. W: Design considerations in more detail. Circulated privately. Available online: http://www.cs.utexas.edu/users/EWD/ewd03xx/EWD302.PDF
[Dijkstra89] Dijkstra, E. W, Feijen,W. H. J., 1988: A Method of Programming. Addison-Wesley.