

Aufgabe 4: Auto-Scrabble

Team: BANDO

Einsendenummer: 00222

1. November 2017

Inhaltsverzeichnis

1	Beantwortung der Fragestellungen	1
1.1	Erste Frage	1
1.2	Zweite Frage	1
2	Lösungsidee	1
3	Beispiele	2
4	Quellcode	2

1 Beantwortung der Fragestellungen

1.1 Erste Frage

Stimmt es, dass TIMO nicht auf einem Kennzeichen stehen kann?

'TIMO' kann nicht auf einem Kennzeichen stehen, da das Wort nur als TI-MO oder TIM-O zerlegt werden kann. Sowohl 'TI' als auch 'TIM' sind keine in der Liste aufgeführten Kürzel.

1.2 Zweite Frage

Gib ein weiteres Wort mit vier Buchstaben an, das nicht auf einem Kennzeichen stehen kann. Findest du sogar ein solches Wort mit drei Buchstaben? Mit zwei?

'OTTO' kann ebenfalls nicht auf einem Kennzeichen stehen, da das Wort nur als OT-TO oder OTT-O zerlegt werden kann. 'OT' und 'OTT' sind erneut keine in der Liste aufgeführten Kürzel.

Ein Wort mit drei Buchstaben, das nicht zerlegt werden kann, lautet 'ICH', da es keine Kürzel gibt, die 'I' oder 'IC' lauten.

Ein Wort mit zwei Buchstaben lautet 'IN', da 'I' kein in der Liste stehendes Kürzel ist.

2 Lösungsidee

Die Hauptidee ist die Umsetzung einer rekursiven Methode, die ein Wort als Parameter erhält. Dabei überprüft sie iterativ alle möglichen Auto-Kennzeichen, die die am Anfang stehenden Buchstaben des erhaltenen Wortes beschreibt. Für jede solche Möglichkeit ruft die Methode sich selbst nochmal auf und übergibt das gleiche Wort abzüglich der ersten Buchstaben, die im gefundenen Auto-Kennzeichen stehen. Lässt sich der Anfang dieses Wortes oder das Wort selbst nicht in einem Kennzeichen darstellen, so gibt die Methode den entsprechenden Wahrheitswert zurück. Ist dieses übergebene Wort ein 'leeres', d.h. stehen keine Buchstaben mehr im übergebenem Wort, so ist eine Lösung gefunden worden, die durch rekursive Rückgaben aufgebaut wird. Sollte dieses Ereignis nicht eintreten, so terminiert das Programm ohne ein Ergebnis.

3 Beispiele

> BIBER

[['B', 'I'], ['B', 'ER']]

> BUNDESWETTBEWERB

[['B', 'U'], ['N', 'D'], ['E', 'S'], ['WE', 'TT'], ['B', 'E'], ['W', 'E'], ['R', 'B']]

> CLINTON

[['C', 'LI'], ['NT', 'ON']]

> DONAUDAMPFSCHIFFFAHRTSKAPITÄNSMÜTZE

"Dieses Wort lässt sich nicht zerlegen."

Grund: 'Ä' kommt weder in einem Kürzel vor, noch darf es als Umlaut im Mittelteil stehen.

> ETHERNET

[['E', 'T'], ['H', 'E'], ['R', 'N'], ['E', 'T']]

> INFORMATIK

[['IN', 'FO'], ['R', 'M'], ['AT', 'IK']]

> LLANFAIRPWLLGWYNGYLLGOGERYCHWYRNDROBWLLLLANTYSILIOGOGOGOCH

[['L', 'L'], ['A', 'N'], ['F', 'AI'], ['R', 'P'], ['W', 'L'], ['L', 'G'], ['W', 'Y'], ['N', 'GY'], ['L', 'L'], ['G', 'O'], ['G', 'E'], ['R', 'Y'], ['C', 'H'], ['W', 'Y'], ['R', 'N'], ['D', 'R'], ['OB', 'W'], ['L', 'L'], ['L', 'L'], ['AN', 'TY'], ['S', 'I'], ['L', 'I'], ['OG', 'O'], ['G', 'O'], ['G', 'O'], ['C', 'H']]

> RINDFLEISCHETIKETTIERUNGSÜBERWACHUNGSAUFGABENÜBERTRAGUNGS-GESETZ

"Dieses Wort lässt sich nicht zerlegen."

Grund: 'Ü' darf als Umlaut nicht im Mittelteil stehen, weshalb es im Kontext eines Kürzels auftreten muss. Es existiert kein Kürzel, das mit 'Ü' beginnt. Vor dem ersten 'Ü' in dem obigen Wort steht ein 'S'. Das einzige Kürzel, in dem die Zeichenkette 'SÜ' vorkommt, ist 'SÜV', was die Bildung des Worts unmöglich macht, da der nachfolgende Buchstabe ein 'B' ist.

> SOFTWARE

[['S', 'O'], ['F', 'T'], ['W', 'A'], ['R', 'E']]

> TRUMP

[['TR', 'U'], ['M', 'P']]

> TSCHÜSS

"Dieses Wort lässt sich nicht zerlegen."

Grund: 'Ü' darf als Umlaut nicht im Mittelteil stehen, weshalb es im Kontext eines Kürzels auftreten muss. Es existiert kein Kürzel, das mit 'Ü' beginnt. Vor dem ersten 'Ü' in dem obigen Wort steht ein 'H'. Es existiert kein Kürzel, in dem die Zeichenkette 'HÜ' vorkommt, was die Bildung des Worts unmöglich macht.

> VERKEHRSWEGEPLANUNGSBESCHLEUNIGUNGSGESETZ

[['V', 'E'], ['R', 'K'], ['E', 'H'], ['R', 'S'], ['W', 'E'], ['G', 'E'], ['P', 'L'], ['A', 'N'], ['UN', 'G'], ['S', 'B'], ['E', 'S'], ['C', 'H'], ['L', 'E'], ['UN', 'I'], ['G', 'U'], ['N', 'G'], ['S', 'G'], ['E', 'S'], ['E', 'TZ']]

4 Quellcode

Der folgende Auszug ist die in der Lösungsidee genannte rekursive Methode:

```

1  # Die rekursive Methode prueft, ob ein Wort als mehere
2  # Nummerschilder geschrieben werden kann
3  def word(wort):
    # Die Liste L wird globalisiert, damit alle
```

```

5  # Rekursionsstufen darauf zugreifen koennen
   global L
7  # Die Liste aller Ortskuerzel wird als i durchiteriert
   for i in k:
9      # Es wird geprueft, ob das aktuelle Ortskuerzel
      # zu dem aktuellen Wort passt
11     if fit(i, wort):
        try:
13         # Eine Liste part mit dem Kuerzel und dem naechsten
            # Buchstaben wird erstellt
15         part = [i, wort[len(i)]]
            # Wenn dieser Buchstabe ein Umlaut ist,
17         if umlaute.count(part[1]) == 1:
            # Dann wird der Boolean False zurueckgegeben
19             return False
            # Wenn das Nummernschild dem uebergebenem Wort entspricht,
21         if part[0] + part[1] == wort:
            # Dann wird das Nummernschild der Liste L hinzugefuegt
23             L.append([part[0], part[1]])
            # Und der Boolean True zurueckgegeben
25             return True
            # Sonst wird das Wort ohne dem Kuerzel und dem
27             # naechsten Buchstaben weitergegeben
            # Wenn True zurueckgegeben wird,
29             elif word(wort[(len(part[0]) + 1):]):
            # Dann wird das Nummernschild der Liste L hinzugefuegt
31                 L.append([part[0], part[1]])
            # Und der Boolean True zurueckgegeben
33                 return True
            # Wenn das Nummernschild mehr Buchstaben besitzt als das Wort
35         except IndexError:
            # Dann wird False zurueckgegeben
37             return False
        try:
39         # Eine Liste part mit dem Kuerzel und den
            # naechsten 2 Buchstaben wird erstellt
41         part = [i, wort[len(i)] + wort[len(i) + 1]]
            # Durchlaufe 0 bis 1 als j
43         for j in range(2):
            # Sollte einer der beiden Buchstaben ein Umlaut sein
45             if umlaute.count(part[1][j]) == 1:
            # Dann wird False zurueckgegeben
47                 return False
            # Wenn das Nummernschild dem uebergebenen Wort entspricht
49             if part[0] + part[1] == wort:
            # Dann wird das Nummernschild der Liste L uebergeben
51                 L.append([part[0], part[1]])
            # Und True zurueckgegeben
53                 return True
            # Sonst wird das restliche Wort weitergegeben
55             # Wenn True zurueckgegeben wird
            elif word(wort[(len(part[0]) + 2):]):
57                 # Dann wird das Nummernschild der Liste L hinzugefuegt
                    L.append([part[0], part[1]])
59                 # und True zurueckgegeben
                    return True
61         # Wenn das Nummernschild laenger als das uebergebene Wort ist
        except IndexError:

```

```
63         # Dann wird False zurueckgegeben
        return False
65     # Wenn keines der Kuerzel passt, dann wird False zurueckgegeben
    return False
67
```
