

# Übung zur Vorlesung “Einführung in die Computerlinguistik und Sprachtechnologie”

Wintersemester 2016/2017, Prof. Dr. Udo Hahn, Sven Büchel  
Übungsblatt 4 vom 25.11.2016  
Abgabe bis 01.12.2016, 23.59 Uhr; per Email (PDF-Format) an  
sven-eric.buechel@uni-jena.de

---

## Aufgabe 1 : Agrammatikalität (2.5 Punkte)

Identifizieren Sie die linguistische Ebene (Phonologie/Graphemik, Morphologie, Syntax, Semantik, Pragmatik), auf der die folgenden Sätze jeweils eine Agrammatikalität aufweisen.

1. \*Der Computer hatte sein Passwort vergessen.
2. \*George Washington plant eine Rede zur Lage im Irak.
3. \*Hölzernhäuser gelten als besonders nachhaltig.
4. \*Er ihm gab 20 Euro.
5. \*Colorless green ideas sleep furiously.

Lösung: Semantik, Pragmatik, Morphologie, Syntax, Semantik.

---

## Aufgabe 2 : Algorithmus “Vokalersetzung” (2,5 Punkte)

Unten sind drei Funktionen angegeben, von denen eine die beiden anderen aufruft. Lösen Sie folgende Aufgaben:

a) 0,5

Welchen Zweck erfüllt die Funktion `vokale`?

**Lösung:** Die Funktion `vokale` testet für ein eingegebenes Wort, wieviele Vokale es enthält und gibt das Ergebnis als Zahl zurück.

b) 0,5

Welchen Zweck erfüllt die Funktion `ersetzeVokale`?

**Lösung:** Die Funktion `ersetzeVokale(wort)` ersetzt in einem eingegebenen Wort alle Vokale durch ein `u`. Das veränderte Wort wird zurück gegeben.

c) 0,5

Welchen Zweck erfüllt die Funktion `aufrufer(wort)`?

**Lösung:** Die Funktion ersetzt in allen Wörtern, die mindestens zwei Vokale enthalten, alle Vokale durch ein `u`.

d) 1

Gegeben seien die Wörter *Glas*, *Donaudampfschiffahrtskapitän*, *Schildkröte*, *Naturschutzpark*, *Uhu*, *jetzt*. Welche Ausgabe hat `aufrufer(wort)` für jedes dieser Wörter? Es reicht die veränderten Wörter anzugeben!

**Lösung:**

Glas, Dunuudumpfschuffuhrtskuputun, Schuldkrutu, Nuturschutzpurk, uhu, jetzt

```
def vokale(↓wort, ↑counter)
  counter ← 0
  for buchstabe in wort
```

```

        if buchstabe ist ein Vokal
            counter ← counter + 1

def ersetzeVokale(↓wort, ↑wort)
    i ← 0
    while i < length(wort)
        if wort[i] ist ein Vokal
            wort[i] ← "u"
        i ← i + 1

def aufrufer(↓wort)
    if vokale(wort) > 1
        wort ← ersetzeVokale(wort)
    print wort

```

---

### Aufgabe 3 : Rekursion (5 Punkte)

Definieren Sie eine rekursive Funktion, die für eine gegebene Liste `zahlen` genau die erste Zahl zurückgibt, die größer als 20 ist (z.B. `zahlen = [3, 1, 32, 5, 40, 7]` → Ausgabe 32). Falls die Liste keine entsprechende Zahl enthält, so soll 20 zurückgegeben werden.

#### Lösung:

```

def groesser20(↓zahlen, ↑zahl)
    if zahlen[0] > 20
        zahl ← zahlen[0]
    else
        if length(zahlen) = 1
            zahl ← 20
        else
            zahl ← groesser20(zahlen[1:])

```