

Übung zur Vorlesung "Computerlinguistik I"

Wintersemester 2016/17, Prof. Dr. Udo Hahn, Sven Büchel

Übungsblatt 2 vom 04.11.2016

Abgabe bis 10.11.2016, 23.59 Uhr; per Email (PDF-Format) an

sven-eric.buechel@uni-jena.de

Aufgabe 1 : Challenge Competitions

- Wozu wird bei Challenges ein Gold Standard Corpus (GSC) verwendet?

Lösung: Zur Evaluation der Systeme; deren Resultate werden mit den Annotationen im GSC verglichen. Teilweise werde auch (separate) GSC zum Training zur Verfügung gestellt.

- Sehen Sie sich die Beschreibungen der Labs¹ (Teilwettbewerbe) der CLEF2016 challenge an - welche beschäftigen sich mit biologischen/medizinischen Problemen?

Lösung:

- CLEF eHealth
 - ImageCLEF
 - LifeCLEF
-

Aufgabe 2 : Programmverstehen

Welche Werte gibt das folgende Programm aus?

```
personen ← ["Anna", "Klaus"]
tiere ← ["Bello", "Maunzele"]
namen ← personen + tiere
x ← namen[2]
if x in personen
    print x + " ist eine Person."
else
    print x + " ist ein Tier."
```

Lösung: Bello ist ein Tier.

¹<http://clef2016.clef-initiative.eu/index.php?page=Pages/labs.html>

Aufgabe 3 : Programmierstrukturen

Im Folgenden sind zwei Aufgaben für kleine Programme gegeben. Formulieren Sie für jedes Problem einen Algorithmus in Form von Pseudocode.

a)

Definieren Sie eine Funktion, die eine Zahl entgegennimmt und ihre Fakultät zurückgibt. Die Fakultät ist das Produkt aller ganzen Zahlen von 1 bis zu dieser Zahl. Die Fakultät von 4 ist also $1 * 2 * 3 * 4 = 24$.

Lösung:

```
def fak(↓zahl, ↑fakultaet)
    i ← 1
    fakultaet ← 1
    while i ≤ zahl
        fakultaet ← i * fakultaet
        i ← i + 1
```

b)

Definieren Sie eine Funktion, die eine Liste (aus Zahlen) durchgeht und überprüft ob die aktuelle Zahl kleiner ist als ihr Vorgänger. Wenn ja sollen beide Zahlen vertauscht werden. Beispielsweise soll $[6, 7, 5]$ zu $[6, 5, 7]$ werden.

Lösung:

```
def sortierschritt(↓zahlen, ↑zahlen)
    i ← 1
    while(i < length(zahlen))
        if(zahlen[i] < zahlen[i-1])
            temp ← zahlen[i]
            zahlen[i] ← zahlen[i-1]
            zahlen[i-1] ← temp
        i ← i + 1
```