

Programmieren I, WS 2011/12

Zusatztutorium Übungen

Aufgabe 1

Welche Werte und Typen geben die folgenden Anweisungen zurück, wenn man sie in den Interpreter eingibt?

Achtung: Mindestens ein Ausdruck führt zu einem Fehler. Welche(r)?

1. 14/3
3. "zz"*3
4. p=3; p=p+4; print q
5. 4.0/0
6. z="2.0"; 18/int(z)
7. "b"+"la"*2

Aufgabe 2

Gegeben sei die folgende Funktion:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
```

```
def distance(wordA, wordB):

    listA = list(wordA)
    listB = list(wordB)
    count = ''

    for char in range(len(listA)):
        if listA[char] != listB[char]:
            count += 1

    print('Die Differenz betraegt ' + count + '!')

    return count
```

a) Was macht die Funktion, und wie geht sie vor? Beschreibe in deinen eigenen Worten!

b) Drei Fehler haben sich eingeschlichen. Welche sind das? Wie kannst du den zweiten Fehler mit einem try/except-Befehl abfangen? Trage ihn oben in die Funktion ein!

Aufgabe 3

Eine häufig wiederkehrende Anwendung in der Computerlinguistik besteht darin, auf die eine oder andere Art und Weise Wörter zu zählen. Dazu muss der Text, in dem wir zählen wollen, aber erst „normalisiert“ werden – das bedeutet, wir müssen alle möglichen Eingabearten auf eine möglichst einheitliche Schreibweise reduzieren (das Wort „der“ könnte beispielsweise als „Der“, „DEr“, „der“, „der“, oder „der!“ im Text vorkommen. Damit wir identische Wörter nicht doppelt zählen, sollten also beispielsweise Satzzeichen herausgefiltert werden.

Wir wollen eine Funktion schreiben, die als ersten Eingabeparameter einen Satz bzw. Text nimmt, und diesen normalisiert. Dazu soll der Satz zeichenweise mit einer Liste erlaubter Zeichen (welche wir als zweiten Eingabeparameter mitgeben) abgeglichen werden. Wenn das untersuchte Zeichen unter den erlaubten Zeichen ist, soll es zu einem neuen String `cleanedText` zugefügt werden. Zurückgegeben werden soll dann der normalisierte Satz. Die Signatur der Funktion ist die folgende:

```
normalizeWords(word, allowedChars) -> str
```

a) Wie gehst du vor? Zeichne ein Flussdiagramm!

b) Jetzt schreibe die Funktion, und versehe sie mit den Kommentaren aus deinem Diagramm!

c) Ändere nun das Ausgabeformat: Was musst du tun, um die normalisierten Wörter direkt in einer Liste zurück zu geben? Wie kannst du das in einer Zeile erreichen? Gib die neue Signatur an!

d) Wenn du die Funktion mit einer for-Schleife implementiert hast: Ändere sie in eine while-Schleife! Hast du eine while-Schleife verwendet, dann ändere sie in eine for-Schleife!

