Funktionsübung

Prof. Dr. Robert Gold

Fakultät Informatik
Technische Hochschule Ingolstadt

Wintersemester 2019/20

Funktionen sind wichtig zur Strukturierung und Wiederverwendung von Programmen. In diesem Dokument finden Sie einige Übungsaufgaben zu Funktionen. Zu jeder Aufgabe soll(en) eine (oder mehrere) Python-Funktion(en) erstellt werden. Außerdem sollen jeweils in einem Hauptprogramm Testaufrufe der Funktionen programmiert werden.

Aufgabe 1 (Aufgabe 1 in Kapitel 2.1)

def third_char(s):	
Parameter:	str s
Rückgabewert:	str : bestehend aus jedem dritten Zeichen in s

Aufgabe 2 (Aufgabe 1 in Kapitel 2.1)

def modulo(a, b):	
Parameter:	int a, b : mit a > 0 und b > 0
Rückgabewert:	int : Rest bei ganzzahliger Division a // b

Es soll solange b von a subtrahiert werden, bis a kleiner als b ist. Die Bedingungen a > 0 und b > 0 müssen nicht überprüft werden.

Aufgabe 3 (Aufgabe 1 in Kapitel 2.1)

def max_three(a, b, c):	
Parameter:	int a, b, c
Rückgabewert:	int : die größte der drei Zahlen

Aufgabe 4 (Aufgabe 1 in Kapitel 2.1)

def inc_second(hours, minutes, seconds):	
Parameter:	int hours, minutes, seconds
Rückgabewert:	(int, int, int) : Tupel bestehend aus der um eine Sekunde erhöhten Uhrzeit

Aufgabe 5 (Aufgabe 4 in Kapitel 2.5)

def is_substring(s, t):	
Parameter:	str s, t
Rückgabewert:	boolean : True, falls die Zeichenkette s in der Zeichenkette t zusammenhängend vorkommt, False, sonst

Aufgabe 6 (Aufgabe 1 in Kapitel 2.6)

def zafo(k):	
Parameter:	int k
Rückgabewert:	int : k-tes Gliedes der Folge 1, 2, 4, 7, 11, 16, 22, …

Die Funktion soll rekursiv sein.

Aufgabe 7

def count_char(s, c):	
Parameter:	str s
	str c : Zeichenkette, die aus genau einem Zeichen besteht
Rückgabewert:	int : Anzahl der Vorkommen von c in s

Die Funktion soll rekursiv sein.

Aufgabe 8 (Kapitel 2.7)

def ggt(a, b):	
Parameter:	int a, b
Rückgabewert:	int : größter gemeinsamer Teiler von |a| und |b|

def kuerze(zaehler, nenner):	
Parameter:	int zaehler, nenner
Rückgabewert:	int, int : zaehler, nenner gekürzt, d.h. geteilt durch den größten gemeinsamen Teiler

def normal(zaehler, nenner):	
Parameter:	int zaehler, nenner
Rückgabewert:	int, int : zaehler, nenner normalisiert

Die Normalisierung soll den Bruch kürzen. Außerdem soll höchstens der Zähler negativ sein. Beispiele:

normal(–1, 2) = –1, 2
normal(–1, –2) = 1, 2
normal(1, –2) = –1, 2

def add(zaehler1, nenner1, zaehler2, nenner2)	
Parameter:	int zaehler1, nenner1, zaehler2, nenner2
Rückgabewert:	int, int : normalisierte Summe der beiden Brüche

def mult(zaehler1, nenner1, zaehler2, nenner2)	
Parameter:	int zaehler1, nenner1, zaehler2, nenner2
Rückgabewert:	int, int : normalisiertes Produkt der beiden Brüche

Erstellen Sie dazu ein Hauptprogramm, in dem der Benutzer die gewünschte Rechnung eingeben kann, z.B. -5/4 + 7/6, und das Programm mit dem Ergebnis antwortet, im Beispiel = –1/12.

In den folgenden Aufgaben sollen keine Funktionen erstellt werden.

Aufgabe 9 (Aufgabe 1 in Kapitel 2.4)

Schreiben Sie ein Skript, das die Datei punkte.csv einliest und auswertet. Die genaue Aufgabenstellung finden Sie in den Folien in Abschnitt 2.4.

Aufgabe 10 (Aufgabe 1 in Kapitel 2.5)

→Aufgabenstellung in den Folien in Abschnitt 2.5.

Aufgabe 11 (Aufgabe 2 in Kapitel 2.5)

→Aufgabenstellung in den Folien in Abschnitt 2.5.

Aufgabe 12

Gegeben sei die folgende Funktion:

```
def what_s_that(s, k):
    if k == 0:
        print(s)
    else:
        what_s_that(s + "0", k - 1)
        what_s_that(s + "1", k - 1)
```

Welche Konsolausgabe hat der Aufruf what_s_that("", 4) ? Beschreiben Sie in einem Satz, was die Funktion ausgibt.