

Einführung in die Programmierung

Prof. Dr. Franziska Matthäus
Prof. Dr. Matthias Kaschube
Dr. Karsten Tolle
Kenan Khauto



Übungsblatt ÜE-02

Ausgabe: 30.10.2024
Abgabe: 09.11.2024
13:00 Uhr

Schleifen

Hinweis:

- *Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein.*
- *Digitale Abgaben, die nicht im Format **.pdf** oder **.txt** für Texte oder **.py** für Code erfolgen, werden nicht bewertet. Bei Abgaben mehrerer Dateien müssen diese als **.zip** zusammengefasst werden.*
- *Achten Sie darauf die Variable **__author__** in allen Quellcode Dateien (.py) korrekt zu setzen (am Anfang des Quellcodes): **__author__ = "<Matr-Nr>, <Nachname>"***
Beispiel: `__author__ = "1234567, Tolle"`
... Leerstellen vor und nach dem „=" und Leerstelle nach dem Komma beachten, sowie keine spitzen Klammern verwenden, die dienen bei der Schreibweise oben zum Definieren von Variablen.
- *Außerdem muss Ihr Name in jeder abgegebenen **.pdf** und **.txt** Datei zu finden sein.*
- *Abgaben der Dokumentation, die per Hand geschrieben und eingescannt werden, sind nur in zuvor abgesprochenen Ausnahmefällen erlaubt.*

Σ 10 Punkte

Aufgabe ÜE-02-1: Schleifen (2 Punkte)

Gegeben sind folgende Code-Stücke. Analysiere und bestimme die Ausgabe.

a.

```
for i in range(1, 4):  
    for j in range(1, 4):  
        for k in range(1, 4):  
            print(f"{i} * {j} * {k} = {i * j * k}")
```

b.

```
count = 3  
while count > 0:  
    for i in range(count):  
        print("Countdown:", count, "-", i)  
    count -= 1
```

c.

```
counter = 1  
while counter <= 5:  
    counter += 1  
    print(counter)
```

d.

```
counter = 1  
while counter <= 5:  
    print(counter)  
    counter += 1
```

Aufgabe ÜE-02-2: ASCII-Kunst - Weihnachtsbaum zeichnen (3 Punkte)

Erstelle einen lauffähigen Python-Code, der einen Weihnachtsbaum aus ASCII-Zeichen in der Konsole darstellt. Das Programm soll die Höhe des Baums (zum Beispiel 5 für einen 5-zeiligen Baum) und die Höhe des Stamms als Eingabe vom Benutzer entgegennehmen und dann einen Weihnachtsbaum ausgeben, der wie folgt aussieht:

```
Gib die Höhe des Baums ein: 5
Gib die Höhe des Stamms ein: 2
  *
 ***
*****
*****
*****
*****
  |||
  |||
```

Dokumentation nicht vergessen!

Aufgabe ÜE-02-3: Python und Mathe (5 Punkte)

Für die Aufgabe ÜE-02-3 ist keine Dokumentation nötig, jedoch soll es im Code sinnvolle Kommentare geben, so dass der Code und Euer Vorgehen verständlich wird! Ebenso soll es im Code für a) und b) entsprechende Testfälle geben.

Löse die folgenden Teilaufgaben (wenn von der Tutorin/Tutor nicht anders gewünscht in zwei separate .py Dateien) :

- a) Die **Fibonacci-Folge** ist eine spezielle Zahlenreihe, bei der jede Zahl (außer den ersten beiden) die Summe der beiden vorherigen Zahlen ist. Sie beginnt normalerweise mit den Zahlen 1 und 1 (es gibt auch Definitionen, die mit „0, 1, 1, ...“ starten). Bei uns:

Fibonacci-Folge: 1, 1, 2, 3, 5, 8, 13, 21, ...

Um den Wert eines beliebigen Elements in der Fibonacci-Folge zu berechnen, können wir eine einfache Formel verwenden:

$$F(n) = F(n - 1) + F(n - 2)$$

Hier bedeutet:

$F(n)$ = die n -te Zahl in der Fibonacci-Folge.

$F(n-1)$ = die Zahl vor $F(n)$ in der Folge.

$F(n-2)$ = die Zahl zwei Positionen vor $F(n)$.

Schreibe ein **Programm**, das den Benutzer nach einer Zahl n fragt und dann die ersten n Zahlen der **Fibonacci-Folge** ausgibt.

1. Die Fibonacci-Folge beginnt mit 1, 1.
2. Jedes Element der Folge ist die Summe der beiden vorherigen Elemente.

Beispiel:

Für $n = 7$ sollte das Programm folgendes ausgeben:

Fibonacci-Folge: 1, 1, 2, 3, 5, 8, 13

- b) Schreibe ein Programm, das den Benutzer nach einer Zahl **n** fragt und dann die folgende mathematische Formel mithilfe von Schleifen berechnet:

$$S = \sum_{k=1}^n \frac{k!}{k+1}$$

Hinweise:

- i. Verwende eine Schleife, um die Fakultät jeder Zahl *k* von 1 bis *n* zu berechnen.
- ii. Berechne für jedes *k* den Bruch $k! / (k + 1)$ und addieren Sie ihn zur Summe *S*.
- iii. Gebe am Ende den berechneten Wert von *S* aus.

Bitte löse die Aufgabe ohne die Verwendung von externen Modulen oder Bibliotheken wie math oder numpy. Alle Berechnungen und Logiken sollten vollständig im Code selbst implementiert werden. Ziel dieser Aufgabe ist es, ein tieferes Verständnis für Schleifen und grundlegende Algorithmen zu entwickeln, ohne auf vorgefertigte Funktionen zurückzugreifen.