

Übungsblatt ÜE-02

Ausgabe: 03.11.2021
Abgabe: 13.11.2021
16:00 Uhr

Schleifen

Hinweis:

- Es sind grundsätzlich Rechenwege anzugeben, es sei denn es findet sich ein expliziter Hinweis, dass dies nicht nötig ist.
- 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 korrekt zu setzen (am Anfang des Quellcodes):
`__author__ = "<Matr-Nr>, <Nachname>"`
- Außerdem muss Ihr Name in jeder abgegebenen **.pdf** und **.txt** Datei zu finden sein.
- Abgaben, die per Hand geschrieben und eingescannt werden, sind **nicht** erlaubt (bzw. geben 0 Punkte und werden nicht korrigiert).

Σ 10 Punkte

Zusatz: Geben Sie den Python Code für die einzelnen Teilaufgaben möglichst in einer einzigen **.py** Datei ab und unterteilen Sie die Aufgabenteile durch entsprechende Kommentare. (Sollte Ihr Tutor ihnen andere Vorgaben gemacht haben, gelten die Vorgaben des Tutors!)

Aufgabe 1 – **while**-Schleife

3 Punkte

- a) Betrachten Sie folgende 3 Schleife. Geben Sie die Ausgabe der jeweiligen Schleife an, und beschreiben Sie warum.

<pre>1 i = 0 2 while i < 5: 3 print(i) 4 i += 1</pre>	<pre>1 i = 0 2 while i < 5: 3 i += 1 4 print(i)</pre>	<pre>1 for i in range(0, 5): 2 print(i)</pre>
--	--	---

- b) Schreiben Sie eine Schleife, die eine Zahl (ungleich 0) so lange halbiert, bis sie nicht mehr dazustellen ist (und an deren Stelle 0.0 verwendet wird). Geben Sie dabei an welche die kleinste Zahl ist und wie viele Schritte dies gebraucht hat.

Aufgabe 2 – for-Schleife

4 Punkte

Erstellen Sie für jede Teilaufgabe eine lauffähigen Python 3.X Code und min. 3 Testfälle als Kommentar unterhalb des Codes, sonst keine Dokumentation notwendig.

- Lesen Sie 2 Ganzzahlen ein und schreiben Sie eine `for`-Schleife, die über beiden Zahlen und den dazwischen liegenden Zahlen die Summe bildet. Z.B. Eingabe 1 und 5, soll $1+2+3+4+5 = 15$ ergeben (die vertauschte Eingabe soll das gleiche Ergebnis liefern, 5 und 1 soll also auch 15 ergeben).
- Erstellen Sie (durch Ausgabe auf der Konsole) ein Schachfeld (8 x 8), indem schwarz als 1 und weiß als 0 codiert ist (Schachfeld soll oben links mit weiß starten).

```
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
```

- Erweitern Sie b) indem die beiden Dimensionen durch eine Eingabe definiert werden. Bei Eingabe von 3 und 9 soll ein Feld (3 x 9) erzeugt werden. Schreiben Sie hierfür eine eigene Funktion, die als Parameter die Dimensionen übergeben bekommt.

Aufgabe 3 – Annäherung Eulersche-Zahl

3 Punkte

Es soll die Eulersche-Zahl angenähert berechnet werden. Die Formel für die Eulersche-Zahl ist:

$$e(n) = \left(1 + \frac{1}{n}\right)^n$$

Diese kann umgeformt werden zu:

$$e(n) = 1 + \sum_{k=1}^n \frac{1}{k!} = 1 + \frac{1}{1} + \frac{1}{1*2} + \frac{1}{1*2*3} + \dots + \frac{1}{1*...*n}$$

Erstellen Sie in Python 3.X eine entsprechende Funktion zur Annäherung an die Eulersche-Zahl und min. 3 Testfälle als Kommentar unterhalb des Codes, sonst keine Dokumentation notwendig.

Die Eulersche-Zahl $e = 2.7182818284 \dots$ ist irrational, hat also wie π unendlich viele Stellen. Schreiben Sie ebenfalls einen Python 3.x Code, der über eine Schleife prüft, wann (also ab welchem n) die ersten oben angegebenen 10 Stellen durch ihre Funktion dargestellt werden.

Zusatzanforderung:

Bei der Umsetzung soll die Fakultät-Berechnung in eine eigene Funktion ausgelagert werden. Die Funktion zur Annäherung an die Eulersche-Zahl soll die Fakultäts-Funktion entsprechend aufrufen. Für die Umsetzung der Fakultät-Berechnung kann der Code aus den Folien zur Vorlesung VE02 (Folie 37) verwendet werden.