

Übungsblatt 9: Python

VORBESPRECHUNG: 05.05.2023

ABGABE: 12.05.2023

Alle Aufgaben sind vollständig zu lösen.

Geben Sie die Übung bitte in Gruppen von 3 oder 4 Personen ab. Erstellen sie dazu ein Team auf ILIAS. Für die Abgabe der Übung selbst laden Sie bitte die Python-Dateien (*.py) hoch. Geben Sie die schriftlichen Aufgaben bitte in Form einer PDF- oder Textdatei (*.pdf oder *.txt) ab. Diese Abgabe wird dann für alle Gruppenmitglieder übernommen.

EIGENE FUNKTIONEN PROGRAMMIEREN

Aufgabe 1

Schreiben Sie eine Funktion, welche über einen formalen Parameter verfügt, welcher dem Radius eines Kreises entspricht. Die Funktion soll die Fläche und den Umfang des Kreises berechnen und die Berechnungen dann mit Hilfe des Schlüsselworts `return` zurückgeben. Rufen Sie die Funktion danach 3-mal mit je einem anderen tatsächlichen Parameter auf und speichern Sie die Rückgabewerte der Funktion in zwei Listen, um diese dann mit Hilfe der Funktion `print` auszugeben.

Tipp: Wenn Sie nicht wissen, wie Sie in einer Funktion mehrere Werte zurückgeben können, betrachten Sie Beispiel 76 im Skript auf Seite 102.

Aufgabe 2

Schreiben Sie eine Funktion `greet`, welche bis zu drei Parameter akzeptiert, jedoch mindestens einen Parameter benötigt, nämlich den Namen einer Person. Unten finden Sie einige Beispiele, welche Ausgaben die Funktion generieren sollten. Passen Sie Ihre Funktion so an, dass die Beispiele in Tabelle 1 funktionieren. Falls Sie Hilfe brauchen, schauen Sie sich Beispiel 73 auf Seite 94 im Skript an.

| Funktionsaufruf | Ausgabe |
|---|--|
| <code>greet("Monika")</code> | Hallo Monika! Guten Morgen! Es ist 20 Grad draussen. |
| <code>greet("Sepp", "Wie geht's denn so?")</code> | Hallo Sepp! Wie geht's denn so? Es ist 20 Grad draussen. |
| <code>greet("Hans", temp=10)</code> | Hallo Hans! Guten Morgen! Es ist 10 Grad draussen. |

Tabelle 1: Ihre Funktion sollte dieselben Ausgaben generieren, wie in dieser Tabelle dargestellt.

Aufgabe 3

Betrachten Sie das untenstehendes Programm. Sie werden bemerken, dass das Programm verschiedene Verantwortungen übernimmt. Ihre Aufgabe ist es, die einzelnen Verantwortungen zu identifizieren und für jede Verantwortung je eine Funktion zu definieren und den Code entsprechend auszulagern. Rufen Sie danach Ihre Funktionen in der richtigen Reihenfolge auf – verwenden Sie wo sinnvoll Parameter und Rückgaben.

```

1 ages = []
2 done = False
3
4 while not done:
5     user_input = int(input("Alter eingeben: (-1 zum Beenden): "))
6     if user_input == -1:
7         done = True
8     else:
9         ages.append(user_input)
10
11 print("Alle Alter aufsteigend sortiert:")
12 ages.sort()
13 for i in range(len(ages)):
14     print("Alter von Student", i, ages[i])
15
16 import statistics
17 print("Mittelwert der Alter: ", round(statistics.mean(ages), 2))

```