

Python – Listen

Im Rahmen der Arbeitsaufträge erweitern Sie folgendes Programm, welches Sie auch auf Ilias finden:

```
1  # Diese Funktion addiert alle Zahlen einer Liste und gibt die Summe zurück
2  def summiereAlleZahlenInListe (liste):
3      zwischensumme = 0
4      for zahl in liste:
5          zwischensumme = zwischensumme + zahl
6      return zwischensumme
7
8
9  # Diese Funktion ermittelt die größte Zahl einer Liste und gibt diese zurück
10 def ermittleGroessteZahlInListe (liste):
11     bisherGroessteZahl = liste[0]
12     for zahl in liste:
13         if zahl > bisherGroessteZahl:
14             bisherGroessteZahl = zahl
15     return bisherGroessteZahl
16
17
18 # Unsere Beispielzahlenliste
19 zahlenliste = [5, 2, 9, 13, 16, 27, 1, 67, 12, 56]
20
21 # Nun testen wir beide Funktionen mit unserer Beispielliste
22 summeDerZahlenliste = summiereAlleZahlenInListe(zahlenliste)
23 print("Die Summe aller Zahlen in der Zahlenliste ist " + str(summeDerZahlenliste))
24
25 groessteZahlInZahlenliste = ermittleGroessteZahlInListe(zahlenliste)
26 print("Die größte Zahl in der Zahlenliste ist " + str(groessteZahlInZahlenliste))
```

Das Programm beinhaltet (für Sie als Beispiel) bereits zwei Funktionen:

- *summiereAlleZahlenInListe* erwartet als Parameter eine Liste und addiert alle Zahlen in dieser Liste zusammen. Die Summe wird als Ergebnis zurückgegeben.
- *ermittleGroessteZahlInListe* erwartet als Parameter ebenfalls eine Liste und ermittelt die größte Zahl in dieser Liste. Diese wird als Ergebnis zurückgegeben.

Zudem beinhaltet das Programm eine Zahlenliste namens *zahlenliste*, welche zehn Zahlen enthält. In den Zeilen 22-26 werden beide Funktionen aufgerufen und *zahlenliste* als Parameter übergeben. Die Funktionen berechnen nun die Summe aller Elemente der Liste bzw. ermitteln dessen größtes Element.

Von Ihnen sind nun weitere Funktionen zu implementieren und mit dieser Zahlenliste und/oder weiteren Zahlenlisten zu testen.

Arbeitsaufträge – Pflichtteil

Info:

Einige der folgenden Aufgaben enthalten Beispiele, welche die Beispielszahlenliste namens *zahlenliste* verwenden, welche auf der vorigen Seite definiert wurde (also die Liste [5,2,9,13,16,27,1,67,12,56]).

Aufgabe 1

Schreiben Sie eine Funktion *multipliziereAlleZahlenInListe (liste)*, welche alle Zahlen der übergebenen Liste multipliziert und das Produkt als Ergebnis zurückgibt.

Beispiel: Der Aufruf *multipliziereAlleZahlenInListe (zahlenliste)* müsste also als Ergebnis 22.756.930.560 zurückgeben (denn $5 \cdot 2 \cdot 9 \cdot 13 \cdot 16 \cdot 27 \cdot 1 \cdot 67 \cdot 12 \cdot 56$ ergibt 22.756.930.560).

Aufgabe 2

Schreiben Sie eine Funktion *ermittleKleinsteZahlInListe (liste)*, welche die kleinste Zahl der übergebenen Liste ermittelt und diese als Ergebnis zurückgibt.

Beispiel: Der Aufruf *ermittleKleinsteZahlInListe (zahlenliste)* müsste also als Ergebnis 1 zurückgeben, denn das ist die kleinste Zahl dieser Liste.

Aufgabe 3

Schreiben Sie eine Funktion *istListeLeer (liste)*, welche ermittelt, ob die übergebene Liste leer ist und entsprechend *True* oder *False* als Ergebnis zurückgibt.

Beispiel: Der Aufruf *istListeLeer (zahlenliste)* müsste also als Ergebnis *False* zurückgeben, denn *zahlenliste* ist keine leere Liste – sie enthält schließlich 10 verschiedene Zahlen.

Aufgabe 4

Schreiben Sie eine Funktion *ermittleIndexDerKleinstenZahl (liste)*, welche den Index (die Position) der kleinsten Zahl in der übergebenen Zahlenliste ermittelt und diesen als Ergebnis zurückgibt.

Beispiel: Der Aufruf *ermittleIndexDerKleinstenZahl(zahlenliste)* müsste also als Ergebnis 6 zurückgeben, weil die kleinste Zahl dieser Liste die 1 ist und diese an Position 6 in der Liste steht.

Aufgabe 5

Schreiben Sie eine Funktion *gibVerdoppelteListeZurueck (liste)*, welche eine neue Liste erstellt und in diese das Doppelte aller Zahlen der übergebenen Liste übernimmt. Diese neue Liste wird als Ergebnis zurückgegeben.

Beispiel: Der Aufruf *gibVerdoppelteListeZurueck (zahlenliste)* müsste also als Ergebnis folgende Liste zurückgeben: [10, 4, 18, 26, 32, 54, 2, 134, 24, 112].

Aufgabe 6

Schreiben Sie eine Funktion *zaehleZweistelligeZahlen (liste)*, welche ermittelt, wie viele Zahlen der übergebenen Liste zweistellig sind (also zwischen 10 und 99 liegen).

Beispiel: Der Aufruf *zaehleZweistelligeZahlen (zahlenliste)* müsste also als Ergebnis 6 zurückgeben, weil diese Liste 6 zweistellige Zahlen enthält.

Aufgabe 7

Schreiben Sie eine Funktion *entferneGeradeZahlen (liste)*, welche aus der übergebenen Liste alle Zahlen löscht, welche gerade sind. Die Liste wird als Ergebnis zurückgegeben.

Beispiel: Der Aufruf *entferneGeradeZahlen (zahlenliste)* müsste als Ergebnis also eine Liste mit folgendem Inhalt zurückgeben: [5,9,13,27,1,67].

Aufgabe 8:

Schreiben Sie eine Funktion *sindAlleZahlenPositiv (liste)*, welche ermittelt, ob alle Zahlen der übergebenen Liste positiv sind.

Beispiel: Der Aufruf *sindAlleZahlenPositiv (zahlenliste)* müsste also als Ergebnis *True* zurückgeben, denn *zahlenliste* enthält keine negativen Zahlen.

Aufgabe 9:

Schreiben Sie eine Funktion *mehrPositiveOderNegativeZahlen (liste)*, welche ermittelt, ob es in der übergebenen Liste mehr positive oder mehr negative Zahlen gibt. Als Ergebnis soll einer der folgenden Texte zurückgegeben (return, nicht print!) werden: „Die Liste hat mehr positive als negative Zahlen“, „Die Liste hat mehr negative als positive Zahlen“ oder „Die Liste hat gleich viele positive und negative Zahlen“.

Beispiel: Der Aufruf *mehrPositiveOderNegativeZahlen (zahlenliste)* müsste als Ergebnis „Die Liste hat mehr positive als negative Zahlen“ zurückgeben – denn *zahlenliste* hat schließlich 10 positive, aber keine negative Zahl.

Arbeitsaufträge – Bonusteil

Info: Die folgenden Aufgaben sind freiwillig. Ihre Bearbeitung geht (je nach Menge der bearbeiteten Aufgaben und der Korrektheit der Lösungen) zusätzlich positiv in die Bewertung mit ein.

Bonusaufgabe 1 (freiwillig!)

Schreiben Sie eine Funktion *enthaeltDuplikate (liste)*, welche ermittelt, ob die übergebene Liste Duplikate enthält (also Zahlen, die mehrfach vorkommen) und entsprechend als Ergebnis *True* oder *False* zurückgibt.

Beispiel: Der Aufruf *enthaeltDuplikate (zahlenliste)* müsste also als Ergebnis *False* zurückgeben, weil diese Liste keine Duplikate enthält.

Bonusaufgabe 2 (freiwillig!)

Schreiben Sie eine Funktion *entferneDuplikate (liste)*, welche alle Duplikate aus der übergebenen Liste entfernt und die (von Duplikaten bereinigte) Liste anschließend als Ergebnis zurückgibt.

Bonusaufgabe 3 (freiwillig!)

Schreiben Sie eine Funktion *habenGemeinsamesElement (liste1, liste2)*, welche ermittelt, ob es in den beiden übergebenen (Zahlen-)Listen mindestens eine Zahl gibt, die in beiden Listen vorkommt.

Beispiel: Der Aufruf *habenGemeinsameElemente ([1,3,3,7], [4,5,6])* soll als Ergebnis also *False* zurückgeben, da es keine Zahl gibt, welche in beiden Listen vorkommt.

Bonusaufgabe 4 (freiwillig!)

Schreiben Sie eine Funktion *ermittleZweitkleinsteZahlInListe (liste)*, welche die zweitkleinste Zahl der übergebenen Liste ermittelt und diese als Ergebnis zurückgibt.

Beispiel: Der Aufruf *ermittleZweitkleinsteZahlInListe (zahlenliste)* müsste als Ergebnis also 2 zurückgeben, denn das ist die zweitkleinste Zahl dieser Liste.

Bonusaufgabe 5 (freiwillig!)

Schreiben Sie eine Funktion *sortiereArray(liste)*, welche die übergebene Zahlenliste aufsteigend sortiert und dann als Ergebnis zurückgibt.

Beispiel: Der Aufruf *sortiereArray(zahlenliste)* müsste als Ergebnis also die Liste [1, 2, 5, 9, 12, 13, 16, 27, 56, 67] zurückgeben.