Aufgaben für Python-Arrays

Aufgabe 1

Schreibe ein Programm, welches alle Elemente (mindestens 10) eines Arrays aus Integern aufsummiert.

Aufgabe 2

Eine Klausuraufgabe lautete: „Schreibe ein Python-Programm, das den Durchschnitt des folgenden Punktespiegels einer Informatikklausur berechnet. Weiterhin soll der Prozentanteil der Klausuren unter 5 Punkten berechnet werden. Du darfst dazu nichts im Kopf addieren.“

Mohamed hat die Aufgabe als einziger richtig gelöst - allerdings ist der Code überhaupt nicht schön, weil ihm noch keine Arrays zur Verfügung standen.

mohamed.py

```
l =
1*0+2*1+2*2+1*3+1*4+3*5+1*6+2*7+3*8+4*9+1*10+2*11+1*12+1*13+1*14+1*15
m = 1+2+2+1+1+3+1+2+3+4+1+2+1+1+1+1
n = str(l/m)
k = 1*0+2*1+2*2+1*3+1*4+3*5+1
p = 1+2+2+1+1+3+1
q = "Der Durchschnitt beträgt:" + n
h = str(k/p)
print(q)
print(h)
```

Löse die Aufgabe, indem du Arrays verwendest. Nutze einmal dafür die dir schon bekannte While-Schleife und einmal die neue For-Schleife, schreibe also zwei Programme.

Tipp: Speichere die Anzahl der Klausuren in einem Array mit 16 Elementen. Der Index repräsentiert dann einfach die Anzahl der Punkte. Welche weiteren Vorteile hat das Verfahren mit den Arrays für die Lehrkraft im Vergleich zu Mohameds Lösung?

Klicke hier für Beispiellösungen

loesung01.py

```
notenspiegel = [1,2,2,1,1,3,1,2,3,4,1,2,1,1,1,1]
num_students = 0

for i in range(16):
    num_students = num_students + notenspiegel[i]
```

loesung02.py

```
notenspiegel = [1,2,2,1,1,3,1,2,3,4,1,2,1,1,1,1]
num_students = 0
num_students_failed = 0
sum_grades = 0

for i in range(16):
    num_students = num_students + notenspiegel[i]
    sum_grades = sum_grades + (notenspiegel[i] * i)

for i in range(4):
    num_students_failed = num_students_failed + notenspiegel[i]

average = round(float(sum_grades/num_students),1)
failed = round(float(num_students_failed/num_students)*100,1)

print("Durchschnitt: ",average)
print("Unter Schnitt: ",failed,"%")
```

Aufgabe 3

Sprachmodelle wie ChatGPT sind in aller Munde. Basis eines Sprachmodells sind immer auch Wortlisten, so wie diese hier:

speechmodel.py

```
verbs = ["geht", "läuft", "spielt", "singt", "schwimmt"]
nouns = ["Peter", "Das Mädchen", "Das Spiel", "Mein Gefühl", "Der Computer"]
parts = ["im Freibad", "fröhlich", "auf den Bäumen", "im Haus", "in der Schule"]
```

Wichtig! Unser „Sprachmodell“ beachtet keine Interpunktion (Satzzeichen). Du kannst das natürlich ergänzen.

1. Erzeuge mit einem Programm aus diesen Wortlisten fünf zufällige Sätze mit einer Methode.
2. Erzeuge mit einem Programm aus diesen Wortlisten fünf zufällige Fragen mit der gleichen Methode wie aus dem Aufgabenteil 1. Die Steuerung, ob Sätze oder Fragen generiert werden, soll über einen boolschen Parameter erfolgen.
3. Beschreibe die Probleme, die bei der Ausgabe auftreten können.

Klicke hier für Beispiellösungen

speechmodel.py

```
import random

verbs = ["geht", "läuft", "spielt", "singt", "schwimmt"]
nouns = ["Peter", "Das Mädchen", "Das Spiel", "Mein Gefühl", "Der
Computer"]
parts = ["im Freibad", "fröhlich", "auf den Bäumen", "im Haus", "in der
Schule"]

# Diese Methode gibt uns ein zufälliges Element aus dem übergebenen
Array zurück
# Durch len(array)-1 kann das Array eine beliebige Länge haben
# Das ist ein klassischer Fall für eine sogenannte "Helfermethode"
def returnRandomElement(array):
    zufallszahl = random.randint(0,len(array)-1)
    return array[zufallszahl]

# Diese Methode baut uns entweder Sätze (isQuestion=False) oder Fragen
(isQuestion=True)
def makeOutput(verbs, nouns, parts, isQuestion):
    if not isQuestion:
        print(returnRandomElement(nouns)+'
'+returnRandomElement(verbs)+' '+returnRandomElement(parts)+'.')
    else:
        print(returnRandomElement(verbs)+'
'+returnRandomElement(nouns)+' '+returnRandomElement(parts)+'?')

# Jetzt können wir sehr einfach Sätze ...
for i in range(5):
    makeOutput(verbs, nouns, parts, False)

print()

# ... und Fragen erstellen.
for i in range(5):
    makeOutput(verbs, nouns, parts, True)
```

Die Ausgaben berücksichtigen keine Groß- und Kleinschreibung und machen semantisch nur selten Sinn. Es handelt sich aber um grammatische Sätze.

From:
https://schule.riecken.de/ - **Unterrichtswiki**

Permanent link:
https://schule.riecken.de/doku.php?id=informatik:algorithmisch:aufgabe:pythonarrays

Last update: **2024/07/17 07:35**