

11_Uebungsaufgaben_Loesungen

0.1 Vorschläge für Übungsaufgaben

Die folgenden Übungsaufgaben sind ausführlicher als die unterwegs vorgeschlagenen Mini-Übungen. Das Ziel ist es hier, nicht nur einzelne Themen direkt auszuprobieren, sondern vielmehr verschiedene Themen zu kombinieren, um so ein verknüpfendes Verständnis der Lerninhalte zu erreichen.

0.1.1 Berechnung des Notendurchschnitts

Diese Aufgabe trainiert Datentypen, elementares Rechnen und while Schleifen. Wir spielen Klausur, und wollen den Notendurchschnitt bestimmen. Weil uns noch einige Sprachmittel fehlen, existiert im folgenden Programm eine Funktion `next_grade()`, die gewissermaßen die Antwort der korrigierenden Person auf unsere Frage "Was ist die nächste Note?" als Black Box bereitstellt. Schreiben Sie ein Programm, dass für eine feste vorgegebene Anzahl von Klausurergebnissen den Notendurchschnitt bestimmt.

```
[4]: #####
# Diesen Teil des Programms können Sie ignorieren
import random

def next_grade():
    return random.randint(1,6)
#####

# Beispiel für die Nutzung: grade = next_grade()

n = 100
sum = 0
i = 1
while i <= n:
    sum = sum + next_grade()
    i = i + 1
mean = sum/n
print("Notendurchschnitt: {:.2f}".format(mean))
```

Notendurchschnitt: 3.21

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

Hierbei ist die Fakultät $k! = 1 \cdot 2 \cdot \dots \cdot k$. Implementieren Sie zunächst die Fakultät mit einer while Schleife. Da unendliche Summen unendlich viele Rechenschritte erfordern, ist es eine nahe-liegende Approximation, die Summation einfach nach n Summanden abzubrechen. Realisieren Sie dies in einer weiteren Schleife um Ihre Fakultätsfunktion herum. Machen Sie sich anschließend in der Dokumentation des Pakets `math` schlau, wie der Betrag einer Zahl bestimmt wird (auf Englisch "absolute value") und wie Sie die Eulersche Zahl als Konstante bekommen. Realisieren Sie damit abschließend für jede Approximationsstufe (für alle Schleifendurchläufe $1, \dots, n$) einen Genauigkeitsvergleich, in dem Sie den relativen Fehler

$$\text{err}_{\text{rel}} = \frac{|\text{value}_k - \text{value}_{\text{math}}|}{|\text{value}_{\text{math}}|}$$

bestimmen.

```
[9]: import math # für den dritten Aufgabenteil

# Fakultätsfunktion
fak = 1
i = 1
k = 5
while i <= k:
    fak = fak * i
    i = i + 1
print("{}! = {}".format(k,fak))

# Approximation der Eulerschen Zahl
n = 10
k = 1
e_approx = 1
print(" Approximation   relevativerFehler") # Aufgabenteil 3
while k <= n:
    fak = 1
    i = 1
    while i <= k:
        fak = fak * i
        i = i + 1
    e_approx = e_approx + 1 / fak
    k = k + 1
    print("{:.15f} {:.15f}".format(e_approx, math.fabs(e_approx - math.e)/math.
    →e))
```

```
5! = 120
Approximation   relevativerFehler
2.000000000000000 0.264241117657115
```

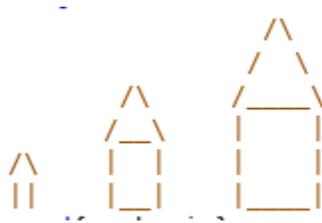
```

2.5000000000000000 0.080301397071394
2.666666666666667 0.018988156876154
2.708333333333333 0.003659846827344
2.716666666666667 0.000594184817582
2.718055555555556 0.000083241149288
2.718253968253968 0.000010249196675
2.718278769841270 0.000001125202598
2.718281525573192 0.000000111425478
2.718281801146385 0.000000010047766

```

0.1.4 Schwäbische Häusebauer

Diese Aufgabe trainiert formatierte Ausgaben und einfache while Schleifen. Für eine beliebige Anzahl n an Etagen soll mit einem Architekturprogramm der Entwurf eines Hauses gezeichnet werden. Für $n = 1, 2, 3$ sieht das Ergebnis so aus:



Realisieren Sie diese Bildschirmausgabe unter Verwendung von drei while Schleifen für das Dach, die Mauern und die Bodenplatte. Jeder Durchlauf durch eine while Schleife zeichnet eine Zeile der entsprechenden Struktur. Gehen Sie langsam vor, zählen Sie vorher anhand einer Skizze (auf Kästchenpapier) ab, an welchen Stellen sich die Zeilen in Abhängigkeit von n verändern, und testen Sie Ihr Programm regelmäßig.

```

[1]: # n = int(input("Geben Sie eine natuerliche Zahl ein:"))
n = 3

# Denkarbeit: relevante Punkte der Häuser, durch Basterei mit Kästchenpapier
breite = 2*(n-1)
wand_hoehe = n-1
dach_hoehe = n-1

# Zeichnen des Daches
while (dach_hoehe>0):
    dach_hoehe=dach_hoehe-1
    print(" *(dach_hoehe+1)+"/"+" "*2*(n-dach_hoehe-2)+"\\") # beachten Sie,
    ↳dass der Backslash escaped werden muss

print("/"+"_"*breite)+"\\") # letzte Dach-Zeile Sonderfall: Dachboden separat

# Mauern
while (wand_hoehe>0):

```

```
wand_hoehe=wand_hoehe-1
print("|"+" "*breite)+"|")

# Fundament
print("|"+"_"*breite)+"|")
```

```
  /\
 /  \
/----\
|      |
|      |
|      |
|----|
```

0.2 Impressum

0.2.1 Programmierkurs Python, Dominik Göddeke <https://www.ians.uni-stuttgart.de>,
Universität Stuttgart

Version vom April 2023

Lizenziert unter der Creative Commons Namensnennung 4.0 International Lizenz



Veröffentlicht auf <https://zoerr.de>, (alle Rechte am Logo vorbehalten)



Gefördert durch die Stiftung Innovation in der Hochschullehre. (alle Rechte am Logo vorbehalten)



Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie.

[]: