

11_Uebungsaufgaben

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.

```
[2]: #####
# Diesen Teil des Programms können Sie ignorieren
import random

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

# Beispiel für die Nutzung: grade = next_grade()

# jetzt sind Sie dran :)
```

0.1.2 Fortschrittsbalken

Diese Aufgabe trainiert formatierte Ausgaben, Strings und while Schleifen. Ergänzen Sie den folgenden Code aus dieser Lerneinheit um eine while-Schleife, d.h. geben Sie alle einzelnen Schritte des Fortschrittsbalkens untereinander aus.

```
[2]: # Lösung
```

0.1.3 Berechnung der Eulerschen Zahl

Diese Aufgabe trainiert elementares Rechnen, while Schleifen und die math Bibliothek. Aus der Schule ist die [Eulersche Zahl](#) bekannt. Sie ist definiert über die folgende endlose Summe:

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

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 abzurechnen. 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

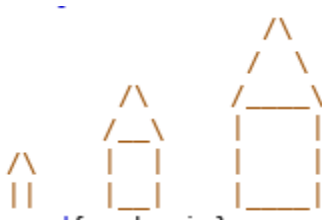
$$\text{err}_{\text{rel}} = \frac{|\text{value}_k - \text{value}_{\text{math}}|}{|\text{value}_{\text{math}}|}$$

bestimmen.

[3]: [# Lösung](#)

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.

[4]: [# Lösung](#)

0.2 Impressum

0.2.1 Programmierkurs Python, Dominik Göttsche <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.

[]: