

**Aufgabenblatt 1 vom 29. Oktober 2021, Abgabe am 5. November 2021, 22:00 Uhr**

---

**Aufgabe 1.1: Grundlagenübung Schleifen**  $\hat{=}$

13 Punkte

*Schleifen*

1. Legen Sie eine neue Datei `schleifen.py` an und programmieren Sie darin die folgenden Schleifen! Kennzeichnen Sie durch Kommentare, welcher Code zu welcher Teilaufgabe gehört!

- Schreiben Sie eine `for`-Schleife, die von 0 bis 10 zählt und die Zahlen auf die Standardausgabe ausgibt.
- Schreiben Sie eine neue Schleife, die in 5er-Schritten von 0 bis 20 zählt.
- ...in 5er-Schritten von 4 bis 34 (einschließlich).
- ...in 2er-Schritten von 11 bis 1 (einschließlich).

2. Schreiben Sie die folgenden `for`-Schleifen als `while`-Schleifen (mit Bedingung und ohne die Verwendung von `range`).

Schreiben Sie Ihre Lösung ebenfalls in die oben angelegte Datei.

```
n = 7
for i in range(n+1):
    print(i)

a = 0
for i in range(2, 20, 3):
    a += i
```

3. Schreiben Sie die `for`-Schleifen aus der vorherigen Teilaufgabe noch einmal als `while`-Schleifen, setzen Sie die Bedingung der Schleife aber nun auf `True` (also `while True: ...`). Verwenden Sie stattdessen im Rumpf der Schleife eine `if`-Abfrage und eine `break`-Anweisung.

4. Die ganzzahlige Division zweier positiver Zahlen  $a$  und  $b$  ( $\lfloor \frac{a}{b} \rfloor$ ) kann man berechnen, indem man zählt, wie oft sich  $b$  von  $a$  abziehen lässt.

Legen Sie zwei Variablen `a` und `b` an und weisen Sie ihnen zunächst die Werte 20 bzw. 3 zu.

Legen Sie eine weitere Variable für das Ergebnis an und verwenden Sie eine `while`-Schleife, um die Division wie oben beschrieben zu implementieren. Verändern Sie `a` und `b` nicht!

Wenn die Division funktioniert, variieren Sie die Werte der beiden Variablen mit Hilfe von zwei `for`-Schleifen. In der äußeren der beiden Schleifen soll  $a$  (mit Schrittgröße 1) rückwärts von 30 bis 1 gehen, in der inneren  $b$  mit der gleichen Schrittgröße von 2 bis 4.

Geben Sie die Ergebnisse in folgendem Format auf die Standardausgabe aus:

```
30 // 2 = 15
30 // 3 = 10
30 // 4 = 7
29 // 2 = 14
...
1 // 4 = 0
```

5. Testen Sie besonders die letzte Teilaufgabe gründlich und geben Sie danach die Datei `schleifen.py` mit allen programmierten Schleifen ab.

## Aufgabe 1.2: Verkaufsautomat

13 Punkte

*Rechnen, Fallunterscheidungen, Schleifen*

In dieser Aufgabe werden Sie einen Verkaufsautomaten programmieren. Der Kunde wählt eines von zwei Produkten, das er dann noch bezahlen muss. Der Verkaufsautomat errechnet das Rückgeld, welches er dann wegen einer technischen Störung leider nur in Kupfermünzen ausgeben kann.

**Hinweis:** Speichern Sie Geldbeträge immer nur als Integer-Werte (in Cent).

1. Legen Sie eine neue Datei `automat.py` an.
2. Begrüßen Sie die Kunden mit einem netten Spruch.
3. Legen Sie zwei Variablen mit den Preisen der beiden Produkte an.
  - Produkt 1 ist ein Standardprodukt und hat deshalb immer denselben Preis.  
Produkt 2 ist ein Tagesgericht und bewegt sich im Bereich zwischen 5,00€ und 19,99€.
  - Nutzen Sie für das Tagesgericht das `random`-Modul:  
<https://docs.python.org/3/library/random.html#random.randrange>  

```
from random import randrange
n = randrange(start, stop)
```
  - Geben Sie alle verfügbaren Produkte mit ihrem Preis in Euro aus, damit sich die Kunden entscheiden können.
4. Erstellen Sie eine Eingabemöglichkeit für die Kunden, mit der sie ein Produkt auswählen können.
  - Überprüfen Sie, ob der eingegebene Text einem der Produkte entspricht, und fragen Sie erneut, falls die Eingabe fehlerhaft war.
  - Geben Sie nach der Eingabe den Preis des Produktes aus.

5. Fordern Sie die Kunden auf, die Ware zu bezahlen.

- Erstellen Sie eine Eingabemöglichkeit, mit der die Kunden den Betrag (in Cent) eingeben können, den sie bezahlen. Versucht ein Kunde, zu wenig zu bezahlen, fragen Sie erneut nach.
- *Optional:* Überprüfen Sie die Eingabe. Ist keine Zahl eingegeben worden, fordern Sie die Kunden erneut auf, bis eine gültige Eingabe erfolgt ist.

6. Geben Sie das Rückgeld aus.

- Errechnen Sie das Rückgeld und geben es in Kupfermünzen zurück!  
Geben Sie an, wie viele 5-, 2- und 1-Cent-Münzen zurückgegeben werden. Es sollen so wenige Münzen wie möglich ausgegeben werden! Gehen Sie davon aus, dass der Automat immer ausreichend viele Kupfermünzen zur Verfügung hat.

7. Bedanken Sie sich bei erfolgreicher Transaktion!

Bitte kommentieren Sie den Code ausführlich und schreiben Sie eine kurze Anleitung für Ihr Programm. Kopieren Sie bitte keinen Code aus dem Internet, auch wir sind in der Lage Google zu benutzen.