

Übungsblatt 03

Aufgabe 1

Gegeben ist eine Liste:

```
punkte = [[40,60],[50,60],[60,60],[70,60],[80,60],[90,60],[100,60],  
[110,60],[120,60]]
```

Die Unterlisten enthalten zwei Zahlen, welche die Koordinaten von Punkten darstellen. Es soll ein Programm erstellt werden, das entscheidet, ob diese Punkte innerhalb eines Kreises liegen. Der Kreis hat die Mittelpunktkoordinate x_k und y_k . Der Durchmesser des Kreises wird in dk gespeichert. Schreiben Sie das Programm in allgemeingültiger Form, sodass beliebige Kreise betrachtet werden können.

Beispielausgabe:

Geben Sie x_k an: 80

Geben Sie y_k an: 50

Geben Sie den Kreisdurchmesser dk an: 60

```
[[40, 60, False], [50, 60, False], [60, 60, True], [70, 60, True], [80,  
60, True], [90, 60, True], [100, 60, True], [110, 60, False], [120, 60,  
False]]
```

```
In [ ]: # TODO Aufgabe 1)
punkte = [[40,60],[50,60],[60,60],[70,60],[80,60],[90,60],[100,60],  
          [110,60],[120,60]]

def punkte_in_kreis(punkte,x,y,D):
    ergebnis = []
    for i in punkte:
        xp = i[0]
        yp = i[1]
        drin = False
        if ((xp-x)**2+(yp-y)**2)<(D/2.)**2:
            drin = True
        ergebnis.append([xp,yp,drin])
    return ergebnis

# Hauptprogramm
xk= int(input("Geben Sie xk an: "))
yk= int(input("Geben Sie yk an: "))
dk= int(input("Geben Sie den Kreisdurchmesser dk an: "))
Ergebnis = punkte_in_kreis(punkte,xk,yk,dk)
print(Ergebnis)
```

```
In [ ]: # Alternative Lösung zu Aufgabe 1) mit List comprehension
punkte = [[40,60],[50,60],[60,60],[70,60],[80,60],[90,60],[100,60],  
          [110,60],[120,60]]
x = int(input("Geben Sie ein xk ein: "))
y = int(input("Geben Sie ein yk ein: "))
d = int(input("Geben Sie einen Kreisdurchmesser an: "))
```

```

ergebnis = [punkt + [True] if (((punkt[0]-x)**2 + (punkt[1]-y)**2)) < (d/2.)**2
            else punkt + [False] for punkt in punkte]
#andere = [punkt.append(True) if (((punkt[0]-x)**2 + (punkt[1]-y)**2)) < (d/2.)**2
#else punkt.append(False) for punkt in punkte]

print(punkte)
print(ergebnis)

```

Aufgabe 2.1

Schreiben Sie ein Programm, dass die Wahrscheinlichkeit dafür berechnet, dass beim Wurf zweier Würfel die Summe der Augenzahlen 7 entspricht. Lösen Sie das Problem mit *list comprehension*. Tipp: nutzen Sie die *built-in function* `sum(iterable)` und zur Ausgabe *f-Strings*

Ausgabe: Die Wahrscheinlichkeit, die Augensumme 7 zu würfeln beträgt:
16.67 %

```

In [ ]: # TODO Aufgabe 2.1)

alleKombinationen = [(x, y) for x in range(1, 7) for y in range(1,7)]
summe7Kombinationen = [x for x in alleKombinationen if sum(x) == 7]
wahrscheinlichkeit = len(summe7Kombinationen)/len(alleKombinationen)
print(f"Die Wahrscheinlichkeit, die Augensumme 7 zu würfeln beträgt: \
      {wahrscheinlichkeit*100:.2f} %")

```

Aufgabe 2.2

Erweitern Sie Ihr Programm um eine Funktion, die obige Wahrscheinlichkeit für eine **beliebige** Summe berechnet und ausgibt. Geben Sie anschließend alle Summen mit ihren Wahrscheinlichkeiten aus.

```

In [ ]: # TODO Aufgabe 2.2)

def wahrscheinlichkeit(summe):
    alleKombinationen = [(x, y) for x in range(1, 7) for y in range(1,7)]
    summeKombinationen = [x for x in alleKombinationen if sum(x) == summe]
    return (len(summeKombinationen)/len(alleKombinationen))

for i in range(14):
    print(f"Summe: {i}, Wahrscheinlichkeit: {wahrscheinlichkeit(i)*100:.2f} %")

```

Aufgabe 3

In einer Buchhandlung findet sich in einem Abrechnungsprogramm in Python eine Liste mit Unterlisten mit folgendem Aufbau:

Bestellnummer	Buchtitel und Autor	Anzahl	Einzelpreis
34587	Learning Python, Mark Lutz	4	40,95 €
98762	Programming Python, Mark Lutz	5	56,80 €

Bestellnummer	Buchtitel und Autor	Anzahl	Einzelpreis
77226	Head First Python, Paul Barry	3	32,95 €

Schreiben Sie ein Programm unter Benutzung von `lambda` und `map`, das als Ergebnis eine Liste mit Zweier-Tupeln liefert. Jedes Tupel besteht aus der Bestellnummer und dem Produkt aus der Anzahl und dem Einzelpreis. Das Produkt soll jedoch um 10,- € erhöht werden, wenn der Bestellwert unter 100,00 € liegt.

Ausgabe: `[('34587', 163.8), ('98762', 284.0), ('77226', 108.85)]`

```
In [ ]: bestellungen = [{"34587", "Learning Python, Mark Lutz", 4, 40.95},
                        ["98762", "Programming Python, Mark Lutz", 5, 56.80],
                        ["77226", "Head First Python, Paul Barry", 3, 32.95]]

# TODO Aufgabe 3)
mindestens = 100
gesamtpreis = list(map(lambda x: x if x[1] >= mindestens else (x[0], x[1] + 10),
                       map(lambda x: (x[0], round(x[2]*x[3], 2)), bestellungen)))
print(gesamtpreis)
```

Aufgabe 4

Erzeugen Sie für einen vorgegebenen Bereich von 1 bis n alle *pythagoreischen Triple* in der Form `p = (a, b, c)`. Für diese muss nach dem Satz des Pythagoras gelten: $a^2 + b^2 = c^2$. Geben Sie die Zahlen sowie die Anzahl der Zahlenpaare im vorgegebenen Bereich aus. Achten Sie darauf, dass keine Doppelten vorkommen, z.B. `(3, 4, 5)` ist das gleiche wie `(4, 3, 5)`

Beispielausgabe:

Geben Sie eine Obergrenze n ein: 20
`[(3, 4, 5), (5, 12, 23), (6, 8, 10), (8, 15, 17), (9, 12, 15)]`
 Zwischen 1 und 20 gibt es 5 pythagoreische Tripel

```
In [ ]: # TODO Aufgabe 4a) mittels Iterationen und Listen
obergrenze = int(input("Geben Sie eine Obergrenze n ein: "))

p = []
for a in range(1, obergrenze):
    for b in range(a, obergrenze):
        for c in range(b, obergrenze):
            if a**2 + b**2 == c**2:
                p.append((a, b, c))
anzahl = len(p)
print(p)
print(f"Zwischen 1 und {obergrenze} gibt es {anzahl} pythagoreische Triple")
```

```
In [ ]: # TODO Aufgabe 4b) mittels List Comprehension
obergrenze = int(input("Geben Sie eine Obergrenze n ein: "))

p = [(a,b,c) for a in range(1, obergrenze)
      for b in range(a, obergrenze)
      for c in range(b, obergrenze)
      if a**2 + b**2 == c**2]
```

```
        if a**2 + b**2 == c**2]
anzahl = len(p)
print(p)
print(f"Zwischen 1 und {obergrenze} gibt es {anzahl} pythagoreische Triple")
```

```
In [ ]: # TODO Aufgabe 4c) mittels Lambda-Formen
obergrenze = int(input("Geben Sie eine Obergrenze n ein: "))

p = list(filter(lambda triple: triple[0]**2 + triple[1]**2 == triple[2]**2,
                [(a, b, c) for a in range(1, obergrenze)
                 for b in range(a, obergrenze)
                 for c in range(b, obergrenze)]))

anzahl = len(p)
print(p)
print(f"Zwischen 1 und {obergrenze} gibt es {anzahl} pythagoreische Triple")
```

```
In [ ]:
```