

Prozedurale Programmierung

Aufgaben 10 Python-Collections

1.

```
import math
import time
t1=time.time()
list=[]
for i in range(1,10000):
    list.append(math.sqrt( i ))
print(time.time()-t1)
print(list)
```

2.

```
# array, list etc sind referenz-parameter
# elementare datentypen, string sind value-parameter
# return (r1,r2,r3) möglich
```

```
def eingabe(list,anzahl):
    for i in range(anzahl):
        x=float(input("x:"))
        list.append(x)

def mittel(list):
    sum=0
    for i in range(len(list)):
        sum+=list[i]
    return sum/len(list)

def fkt(list,mw):
    kl=gr=gl=0
    for i in range(len(list)):
        if mw<list[i]:
            kl=kl+1
        elif mw>list[i]:gr=gr+1
        else: gl=gl+1
    return (kl,gr,gl)
```

```
anzahl=int(input("Anzahl:"))
list=[]
```

```

eingabe(list,anzahl)
print(list)
print(mittel(list))
(r1,r2,r3)=fkt(list,mittel(list))
print(r1,r2,r3)

```

3.

#Ausgabe der Hauptstadt ausgewaehlter suedamerikanischer Laender

```

landHauptstadtDict={"Peru":"Lima","Argentinien":"Buenos Aires", "Chile":"Santiago de Chile", "Brasilien":"Brasilia", "Uruguay":"Montevideo", "Venezuela":"Caracas", "Ecuador":"Quito"}

```

```

print(landHauptstadtDict)
keys = landHauptstadtDict.keys()
print(keys)
for key in keys:
    print(key, ' ', landHauptstadtDict[key])
# for x, y in thisdict.items():
#     print(x, y)

```

```

key=input()
while key!="Ende":
    if key in landHauptstadtDict:
        print(landHauptstadtDict[key])
    else:
        print("leider nicht enthalten")
    key=input()

```

4. (Zusatz)

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

```

```

class Station:
    def __init__(self, id, loc="", sensors=[]):
        self.id = id
        self.location = loc
        self.sensors = sensors

    def add_sensor(self, sensor):
        self.sensors.append(sensor)

```

```

class Management:
    def __init__(self, stations=[]):

```

```

        self.stations = stations

    def add_station(self, station):
        self.stations.append(station)

    def remove_station(self, id):
        for i in range(len(self.stations)):
            if self.stations[i].id == id:
                del self.stations[i]
                break

    def find_stations(self, loc):
        return [s for s in self.stations if s.location == loc]
        # oder mit initialisierung, for, append

    def print(self):
        for s in self.stations:
            print("-"*15)
            print(f"Station #{s.id}\n\tLocation:\t{s.location}\n\tSensors:\t{s.sensors}")
            print()

if __name__ == "__main__":
    manager = Management()

    manager.add_station(Station(10, "Freiberg", ["Temp", "Hum"]))
    manager.add_station(Station(11, "Freiberg", ["Temp", "Hum", "Press"]))
    manager.add_station(Station(20, "Dresden", ["Temp", "Press"]))
    manager.add_station(Station(22, "Dresden", ["Temp", "Hum"]))
    manager.add_station(Station(30, "Leipzig", ["Temp"]))
    manager.add_station(Station(40, "Chemnitz", ["Temp", "Hum", "Press"]))

    manager.print()

    fg = manager.find_stations("Freiberg")
    print("Stations in Freiberg: ", end="")
    for s in fg: print(s.id, end=" ", " ")
    print()

    manager.remove_station(11)
    manager.print()

```