

10B - Weitere Datenstrukturen

Programmieren für Naturwissenschaften FS 2023

Gruppe: Sofia Kessler, Florian Mohaupt, Lukas Batschelet

Aufgabe 1: 2D-Liste

Aufgabe

Definieren Sie die folgende Matrizen mithilfe von Listen:

$$A = \begin{pmatrix} 11 & 2 & 3 \\ 4 & 2 & 6 \\ -11 & 24 & -1 \end{pmatrix} \quad B = \begin{pmatrix} 9 & 21 & 5 \\ 1 & -3 & 3 \\ 9 & -8 & 2 \end{pmatrix}$$

Addieren Sie die beiden Matrizen und geben Sie das Resultat aus. Bei der Addition von Matrizen wird jeder Eintrag komponentenweise addiert, z.B.:

$$\begin{pmatrix} a_{1|1} & a_{1|2} \\ a_{2|1} & a_{2|2} \end{pmatrix} + \begin{pmatrix} b_{1|1} & b_{1|2} \\ b_{2|1} & b_{2|2} \end{pmatrix} = \begin{pmatrix} a_{1|1} + b_{1|1} & a_{1|2} + b_{1|2} \\ a_{2|1} + b_{2|1} & a_{2|2} + b_{2|2} \end{pmatrix}$$

Tipp: Verwenden Sie einen Loop für die Berechnung.

Mögliche Lösung

```
# Definition der Matrizen A und B
A = [[11, 2, 3],
      [4, 2, 6],
      [-11, 24, -1]]

B = [[9, 21, 5],
      [1, -3, 3],
      [9, -8, 2]]

# Initialisierung der Ergebnismatrix C mit Nullen
C = [[0, 0, 0],
      [0, 0, 0],
      [0, 0, 0]]

# Addition der Matrizen A und B, Speicherung in Matrix C
for i in range(len(A)):
    for j in range(len(A[0])):
        C[i][j] = A[i][j] + B[i][j]

# Ausgabe der Ergebnismatrix C
print("Die addierte Matrix C ist:")
for row in C:
    print(row)
```

Aufgabe 2: 2D-Listen

Aufgabe

Gegeben ist folgendes Code-Fragment:

```
animals = {'rabbit', 'tiger', 'dog', 'bird', 'cat', 'zebra', 'koalas'}
animals.update(['elephant', 'deer', 'shark', 'giraffe', 'cat'])
animals.discard('koalas')
carnivores = {'tiger', 'dog', 'bird', 'cat', 'shark'}

print(animals)
print(animals.difference(carnivores))
```

Gehen Sie den Code durch und geben Sie die Ausgabe des Programms an. Verwenden Sie dabei keinen Computer / Laptop.

Lösung

1. Die Menge `animals` enthält anfangs die Elemente: {'rabbit', 'tiger', 'dog', 'bird', 'cat', 'zebra', 'koalas'}.
2. Durch die `update`-Methode werden die Elemente 'elephant', 'deer', 'shark', 'giraffe', und 'cat' zu `animals` hinzugefügt. 'Cat' ist bereits vorhanden, daher bleibt es ein einzigartiges Element. Die Menge `animals` sieht nun so aus: {'rabbit', 'tiger', 'dog', 'bird', 'cat', 'zebra', 'koalas', 'elephant', 'deer', 'shark', 'giraffe'}.
3. Mit der `discard`-Methode wird 'koalas' aus der Menge `animals` entfernt. Danach enthält `animals`: {'rabbit', 'tiger', 'dog', 'bird', 'cat', 'zebra', 'elephant', 'deer', 'shark', 'giraffe'}.
4. Die Menge `carnivores` enthält die Elemente: {'tiger', 'dog', 'bird', 'cat', 'shark'}.
5. Die `difference`-Methode zeigt die Elemente in `animals`, die nicht in `carnivores` sind. Das wären: {'zebra', 'elephant', 'deer', 'rabbit', 'giraffe'}.

Daher würden die Ausgaben des Programms wie folgt aussehen:

- `Animals:` zeigt die endgültige Menge `animals`.
 - `Animals:` {'cat', 'giraffe', 'elephant', 'dog', 'bird', 'shark', 'zebra', 'deer', 'rabbit', 'tiger'}
- `Difference:` zeigt die Elemente, die nur in `animals` aber nicht in `carnivores` sind.
 - `Difference:` {'giraffe', 'deer', 'elephant', 'rabbit', 'zebra'}

Aufgabe 3: Wörterbuch

Aufgabe

Definieren Sie ein Wörterbuch, welches die folgenden Informationen enthält:

ID	Name
1	Samuel Meier
2	Hans Peterson
3	Klara Schmidt
4	Theresa Gerber

Schreiben Sie ein Programm, welches das Abfragen und das Registrieren von Kunden ermöglicht. Dabei soll die Kunden-ID automatisch generiert werden. Dies könnte zum Beispiel so aussehen:

```
Geben Sie den Namen des Kunden ein: Klara Schmidt
ID: 4, Name: Klara Schmidt
Weitere Abfrage? (y/n) y
Geben Sie den Namen des Kunden ein: Sarah Grün
Dieser Kunde ist nicht registriert, möchten Sie ihn aufnehmen? (y/n) y
Kunde wurde hinzugefügt.
```

Mögliche Lösung

```
# Wörterbuch zur Speicherung der Kundeninformationen
kunden_dict = {
    1: 'Samuel Meier',
    2: 'Hans Peterson',
    3: 'Klara Schmidt',
    4: 'Theresa Gerber'
}

# Funktion zur automatischen Generierung einer Kunden-ID
def generiere_kunden_id(kunden_dict):
    return max(kunden_dict.keys()) + 1

# Hauptprogramm
while True:
    name = input("Geben Sie den Namen des Kunden ein: ")
    # Überprüfung, ob der Kunde bereits im Wörterbuch ist
    if name in kunden_dict.values():
        kunde_id = [k for k, v in kunden_dict.items() if v == name][0]
        print(f"ID: {kunde_id}, Name: {name}")
    else:
        print("Dieser Kunde ist nicht registriert, möchten Sie ihn aufnehmen? (y/n)")
        antwort = input()
        if antwort.lower() == 'y':
            neue_id = generiere_kunden_id(kunden_dict)
            kunden_dict[neue_id] = name
            print(f"Kunde wurde hinzugefügt. ID: {neue_id}, Name: {name}")
    # Abfrage für weitere Eingabe
    weitere_abfrage = input("Weitere Abfrage? (y/n) ")
    if weitere_abfrage.lower() != 'y':
        break
```