Heinrich Heine Universität Düsseldorf

Mathematisches Institut
Prof. Dr. Achim Schädle
Melinda Hagedorn

17.10.2024

Computergestützte Mathematik zur linearen Algebra – 2. Übungsblatt

Aufgabe 5: (*Kopieren*)

Führen Sie zunächst die Befehle `from copy import copy` und `from copy import deepcopy` aus. Erstellen Sie nun folgende Objekte des Typs `list`:

$$\texttt{A} = [1,[2,3]], \quad \texttt{B}_1 = \texttt{A}, \quad \texttt{B}_2 = \texttt{A[:]}, \quad \texttt{B}_3 = \texttt{copy(A)}, \quad \texttt{B}_4 = \texttt{deepcopy(A)}$$

Wenden Sie nun die `id` Funktion auf `A`, `A[0]`, `A[1]`, sowie $\texttt{B}_i$, $\texttt{B}_i\texttt{[0]}$, $\texttt{B}_i\texttt{[1]}$ für $i = 1,2,3,4$ an und vergleichen Sie die Rückgabewerte. Was fällt Ihnen auf? Erklären Sie die Unterschiede.

Aufgabe 6: (*Einführung Funktionen: Quader*)

(a) Schreiben Sie eine Funktion `Quader(a,b,c)`, welche das Volumen sowie die Oberfläche eines Quaders mit den Kantenlängen `a, b` und `c` berechnet. Geben Sie eine aussagekräftige Fehlermeldung aus, falls mindestens eine der Kantenlänge kleiner als 0 ist.

(b) Erweitern Sie Ihre Funktion um die Default-Werte `a = b = c = 1`.

(c) Testen Sie Ihre Funktion mit den Eingabewerten (`a=-1,b=2,c=3`), (`a=-1,b=-2,c=3`), (`a=1,b=2,c=3`), (`a=3,b=2,c=1`), (`a=3,b=c=1`), (`a=c=1,b=3`) und (`a=b=c=1`) (Die Klammern dienen hier nur der Lesbarkeit).

Aufgabe 7: (*Rekursionen verstehen*)

Gegeben ist folgender Python-Code, der auf https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/ zu finden ist:

```
def TowerOfHanoi(n, from_rod, to_rod, aux_rod):
    if n == 0:
        return
    TowerOfHanoi(n-1, from_rod, aux_rod, to_rod)
    print("Move disk", n, "from rod", from_rod, "to rod", to_rod)
    TowerOfHanoi(n-1, aux_rod, to_rod, from_rod)
# Driver code
N = 3
# A, C, B are the name of rods
TowerOfHanoi(N, "A", "C", "B")
```

Die Funktion `TowerOfHanoi` löst das mathematische Knobelspiel „Türme von Hanoi“, in dem es drei Stäbe `from_rod`, `aux_rod` und `to_rod` sowie `n` Scheiben gibt, die von `from_rod` nach `to_rod` transportiert werden.

(a) Verstehen und erläutern Sie die Ausgabe. Hier kann ein Blick in die Spielregeln (z.B. bei Wikipedia) nicht schaden. Es gibt auch zahlreiche Online-Versionen des Spiels, die helfen können.

(b) Verstehen Sie den Algorithmus. Schreiben Sie dazu die einzelnen Schritte auf Papier auf, die bei Aufruf des Programms für `N=1`, `N=2` und `N=3` ausgeführt werden.

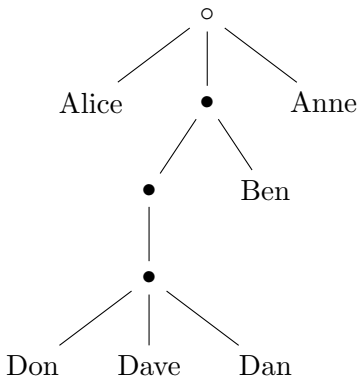
b.w.

Aufgabe 8: (*Rekursionen implementieren*)

Mit geschachtelten Listen kann man in PYTHON baumartige Datenstrukturen mit Wurzel ∘ und Verzweigungen • erzeugen. So könnte man nebenstehenden Baum als

```
[Alice, [[[Don, Dave, Dan]], Ben], Anne]
```

abspeichern.



Schreiben Sie eine rekursive Funktion `zaehle(gl)`, die in der geschachtelten Liste `gl`, die die Blätter, in diesem Fall die Namen, zählt.

Gehen Sie dazu wie folgt vor:

Für eine geschachtelte Liste wird ein Element `el` der Liste entnommen. Dann werden durch einen rekursiven Aufruf von `zaehle(el)` und `zaehle(rest_der_liste)` die Blätter von `el` und die von `rest_der_liste` gezählt. Hier ist `rest_der_liste` der Rest der geschachtelten Liste aus der das Element `el` entfernt wurde.

Hinweis: Wenn `el` ein Eintrag einer Liste ist, können Sie mit `isinstance(el, list)` testen, ob `el` eine Liste ist. `list` ist der Typ des Listenobjekts. Die leere Liste `[ ]`, die den Baum bezeichnet der nur aus der Wurzel besteht, ist `False`, und hat kein Blatt. Eine Liste mit mindestens einem Eintrag ist `True`.

Besprechung in den Übungen vom 21.10.2024 bis 25.10.2024.