

Übungsblatt 4

Aufgabe 1

Implementieren Sie, gestützt auf *Dictionaries*, ein Programm, das einen eingegebenen Ziffernstring in Morsezeichen übersetzt.

Hinweis: Die Morsezeichen der Ziffern lauten:

1 = *---- 2 = **--- 3 = ***-- 4 = ****- 5 = *****
6 = -**** 7 = --*** 8 = ---** 9 = ----* 0 = -----

In [1]: `# TODO Aufgabe 1`

Aufgabe 2

Schreiben Sie einen Generator, der die Fibonacci-Zahlen generiert. Zur Erinnerung: die n -te Fibonacci-Zahl errechnet sich nach:

$$F(n) = F(n-1) + F(n-2) \text{ mit } F(0) = 0 \text{ und } F(1) = 1$$

Beispiel: $F(50) = 12\,586\,269\,025$

In [2]: `# TODO Aufgabe 2`

Aufgabe 3

Schreiben Sie einen Generator `frange(...)` analog zu `range(...)`, der jedoch mit float-Zahlen funktioniert.

Bei folgenden Aufrufen:

```
for i in frange(5.6):
    print(i, end=", ")
print()
for i in frange(0.3, 5.6):
    print(i, end=", ")
print()
for i in frange(0.3, 5.6, 0.8):
    print(i, end=", ")
print()
for i in frange(3.5, 1.2, -0.7):
    print(i, end = ", ")
```

sollte die Ausgabe so aussehen:

```
0, 1, 2, 3, 4, 5,
0.3, 1.3, 2.3, 3.3, 4.3, 5.3,
0.3, 1.1, 1.9000000000000001, 2.7, 3.5, 4.3, 5.1,
3.5, 2.8, 2.0999999999999996, 1.3999999999999997,
```

In []: *# TODO Aufgabe 3*

```
for i in frange(5.6):
    print(i, end=", ")
print()
for i in frange(0.3, 5.6):
    print(i, end=", ")
print()
for i in frange(0.3, 5.6, 0.8):
    print(i, end=", ")
print()
for i in frange(3.5, 1.2, -0.7):
    print(i, end = ", ")
```

Aufgabe 4

In einem Tanzkurs befinden sich folgende Personengruppen:

Vier Damen ("ABCD"), vier Herren ("abcd") und ein Tanzlehrer ("X").

Berechnen Sie mithilfe von Mengen-Objekten und Tupeln die Menge aller denkbaren Tanzpaare. Ein Tanzpaar besteht aus einer Dame und einem Herrn. Tanzlehrer können sowohl die Rolle einer Dame als auch die Rolle eines Herrn übernehmen. *Tipp: Generatorausdrücke eignen sich zur Beschreibung einer Menge.*

Beispielausgabe:

```

('X', 'c') ('C', 'X') ('c', 'B') ('X', 'D') ('X', 'B') ('d', 'D') ('C',
'd') ('C', 'c') ('a', 'B') ('A', 'd') ('X', 'a') ('b', 'B') ('a', 'D')
('c', 'D') ('C', 'a') ('X', 'b') ('b', 'D') ('C', 'b') ('A', 'c') ('A',
'X') ('X', 'd') ('A', 'a') ('d', 'B') ('A', 'b')

```

In [4]: `# TODO Aufgabe 4`

Aufgabe 5

Schreiben Sie ein Programm, das das Mischen der 32 Karten eines Skatspiels simuliert. Jede Karte wird durch ein Tupel der Form `(farbe, wert)` dargestellt. Farben sind: Kreuz, Pik, Herz und Karo. Werte sind: Ass, König, Dame, Bube, 10, 9, 8, 7.

In dem Programm wird zunächst mithilfe eines geeigneten Listendisplays (z.B. `list comprehension`) eine Liste mit den 32 Karten-Tupeln erstellt. Durch wiederholtes (zufällig zwischen 20 und 40 mal) paarweises Vertauschen zweier zufällig ausgewählter Listenelemente wird gemischt.

Nutzen Sie für die Erzeugung von Zufallszahlen die Methode `randint(start, end)` aus dem Modul `random`. Diese erzeugt eine zufällige Integer `n`, wobei gilt:
 $start \leq n \leq end$.

In [5]: `# TODO Aufgabe 5`
`import random`

Aufgabe 6

In dieser Aufgabe beschäftigen Sie sich nochmals mit dem Datentyp `dictionary`. Zur Übung sollen Sie Daten, die in einer Liste abgelegt sind, in ein Dictionary umwandeln.

Aufgabe 6.1

Die gegebene Liste `actors` besteht aus jeweils fünf Unterlisten mit jeweils fünf Tupeln unterschiedlicher Länge. Jedes dieser Tupel enthält Einträge vom Typ `String`.

Wandeln Sie die Liste in ein Dictionary folgender Form um:

```

{0: {'salutation': 'Mr.',
    'givenName': 'Peter',
    'familyName': 'Dinklage',
    'birthday': '06/11/1969',
    'GoT': ['Tyrion', 'Lannister']}},
1: {'salutation': 'Mrs.',
    'givenName': 'Michelle',
    ... }
...}

```

In [6]: `actors = [{"salutation", "Mr."}, ("givenName", "Peter"),
 ("familyName", "Dinklage"), ("birthday", "06/11/1969"),`

```

        ("GoT", "Lannister", "Tyrion"]],
[("salutation", "Mrs."), ("givenName", "Michelle"),
 ("familyName", "Fairley"), ("birthday", "01/17/1964"),
 ("GoT", "Stark", "Catelyn"]],
[("salutation", "Mr."), ("givenName", "Nicolaj"),
 ("familyName", "Coster-Waldau"), ("birthday", "07/27/1970"),
 ("GoT", "Lannister", "Jamie"]],
[("salutation", "Ms."), ("givenName", "Maisie"),
 ("familyName", "Williams"), ("birthday", "04/17/1997"),
 ("GoT", "Stark", "Arya"]],
[("salutation", "Mrs."), ("givenName", "Emilia"),
 ("familyName", "Clarke"), ("birthday", "05/01/1987"),
 ("GoT", "Targaryen", "Daenerys"]]]

```

TODO Aufgabe 6.1)

Aufgabe 6.2 (für Freaks)

Erstellen Sie nun durch geschickte Umformung ein neues Dictionary. Bei diesem dient der Familienname des gespielten Charakters als Schlüssel des äußeren Dictionary. Der Wert zu jedem Schlüssel ist wiederum ein Dictionary. In diesem (inneren) Dictionary werden nun die Vornamen der Charaktere als Schlüssel verwendet. Die Werte in den inneren Dictionaries sind dann die Daten des Schauspielers des jeweiligen Charakters. Das resultierende Dictionary hat also die folgende beispielhafte Struktur:

```

{'Lannister': { 'Tyrion': {'salutation': 'Mr.',
                          'givenName': 'Peter',
                          'familyName': 'Dinklage',
                          'birthday': '06/11/1969'},
               'Jamie': {'salutation': 'Mr.',
                          'givenName': 'Nicolaj',
                          'familyName': 'Coster-Waldau',
                          'birthday': '07/27/1970'}}},
'Stark': {    'Catelyn': {'salutation': 'Mrs.',
                          ...}}

...}

```

Geben Sie außerdem aus, wie viele Mitglieder die fiktive Familie "Stark" nach Ihrem Dictionary hat:

Die Familie Stark hat laut diesem Datensatz 2 Mitglieder

In [7]: *# TODO Aufgabe 6.2)*
Zelle mit actors-Liste muss vorher ausgeführt worden sein, damit diese im Work