

## Übungsblatt 3

### Aufgabe 1

- 1: itertools (eine Sammlung von nützlichen Python Befehlen) soll importiert werden um sie später benutzen zu können.
- 2: Alle mit # versehenen Zeilen sind keine Befehle in dem Sinn sondern lediglich Erklärungen. Eine Grundmenge soll also erstellt werden.
- 3: Diese Grundmenge sind die Augenzahlen von 1-6.
- 4: Wiederum eine Erklärung.
- 5: Der Ergebnisraum „Omega“ =  $\Omega$  soll das „set“ (eine gelungene Übersetzung fällt mir leider nicht ein, aber in die Richtung von „die Zusammensetzung“ in Form von „Liste“) vom kartesischen Produkt der Augenzahlen.
- 6: Dies soll berechnet werden.
- 7-9: Erklärungen, dass der Befehl „list“ statt „set“ den Output sogleich sortieren würde.
- 12: Der Anfangswert für „günstig“ ist bei 0.
- 13: Für die möglichen Ereignisse soll auf den Ergebnisraum  $\Omega$  zurückgegriffen werden.
- 14: Es wird definiert: Bezüglich Augenzahl 1 und Augenzahl 2 in  $\Omega$ :
- 15: Falls deren Summe gleich 10 ist:
- 16: dann wird der Wert für „günstig“ um +1 erhöht.
- 19 & 20: Für die endgültige Ausgabe sollen die günstigen durch die möglichen Ereignisse dividiert werden.

### Aufgabe 2

Für die zweite Aufgabe wurden Vorher/Nachher Screenshots zur besseren Veranschaulichung gemacht:

Vorher:

```
1 import itertools
2 # 1. Grundmenge erstellen:
3 Augenzahlen = {1, 2, 3, 4, 5, 6}
4 # itertools.product liefert das kartesische Produkt:
5 Q = set(itertools.product(Augenzahlen, Augenzahlen))
6 print(Q)
7 # Beachte, dass Q als Menge nicht sortiert ist.
8 # Statt "set" kann auch "list" geschrieben werden:
9 # dann wird die Menge schon sortiert.
10
11 # 2. Anzahl günstige und Mögliche bestimmen:
12 günstig = 0
13 möglich = len(Q)
14 for Augenzahl1, Augenzahl2 in Q:
15     if Augenzahl1 + Augenzahl2 == 10:
16         günstig = günstig + 1
17
18 # 3. Ausgabe:
19 print(f"{günstig} von {möglich}")
20 print(f"p = {günstig/möglich}")
21
```

(3, 4), (4, 3), (3, 1), (5, 4), (4, 6), (5, 1), (2, 2), (1, 6), (2, 5), (1, 3), (6, 2), (6, 5), (4, 2), (4, 5), (3, 3), (5, 6), (3, 6), (5, 3), (2, 4), (1, 2), (2, 1), (1, 5), (6, 1), (6, 4), (3, 2), (4, 1), (3, 5), (5, 2), (4, 4), (5, 5), (1, 1), (1, 4), (2, 3), (2, 6), (6, 6), (6, 3)

3 von 36  
p = 0.08333333333333333

Nachher:

```
1 import itertools
2 # 1. Grundmenge erstellen:
3 Augenzahlen = {1, 2, 3, 4, 5, 6}
4 # itertools.product liefert das kartesische Produkt:
5 Q = list(itertools.product(Augenzahlen, Augenzahlen))
6 print(Q)
7 # Beachte, dass Q als Menge nicht sortiert ist.
8 # Statt "set" kann auch "list" geschrieben werden:
9 # dann wird die Menge schon sortiert.
10
11 # 2. Anzahl günstige und Mögliche bestimmen:
12 günstig = 0
13 möglich = len(Q)
14 for Augenzahl1, Augenzahl2 in Q:
15     if Augenzahl1 + Augenzahl2 == 10:
16         günstig = günstig + 1
17
18 # 3. Ausgabe:
19 print(f"{günstig} von {möglich}")
20 print(f"p = {günstig/möglich}")
21
```

((1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5), (3, 6), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (5, 1), (5, 2), (5, 3), (5, 4), (5, 5), (5, 6), (6, 1), (6, 2), (6, 3), (6, 4), (6, 5), (6, 6))

3 von 36  
p = 0.08333333333333333

Wir sehen also, dass der Befehl „list“ insofern praktischer als „set“ ist, da er die kartesischen Produkte bereits nach aufsteigender Augenzahl ordnet.

### Aufgabe 3

Hier war mir die Aufgabenstellung nicht 100%ig klar, ich hoffe ich habe es richtig verstanden: In Zeile 5 & 14 habe ich um eine dritte Augenzahl erweitert und in Zeile 15 die Lösung vom letzten Übungsblatt importiert. Im Folgenden findet sich ein Screenshot.

```

1 import itertools
2 # 1. Grundmenge erstellen:
3 Augenzahlen = {1, 2, 3, 4, 5, 6}
4 # itertools.product liefert das kartesische Produkt:
5 Q = list(itertools.product(Augenzahlen, Augenzahlen,
6                             Augenzahlen))
7 print(Q)
8 # Beachte, dass Q als Menge nicht sortiert ist.
9 # Statt "set" kann auch "list" geschrieben werden:
10 # dann wird die Menge schon sortiert.
11
12 # 2. Anzahl Günstige und Mögliche bestimmen:
13 günstig = 0
14 möglich = len(Q)
15 for Augenzahl1, Augenzahl2, Augenzahl3 in Q:
16     if Augenzahl1 <= Augenzahl2 <= Augenzahl3:
17         günstig = günstig + 1
18
19 # 3. Ausgabe:
20 print(f"{günstig} von {möglich}")
21 print(f"p = {günstig/möglich}")

```

), (2, 6, 5), (2, 6, 6), (3, 1, 1), (3, 1, 2), (3, 1, 3), (3, 1, 4), (3, 1, 5), (3, 1, 6), (3, 2, 1), (3, 2, 2), (3, 2, 3), (3, 2, 4), (3, 2, 5), (3, 2, 6), (3, 3, 1), (3, 3, 2), (3, 3, 3), (3, 3, 4), (3, 3, 5), (3, 3, 6), (3, 4, 1), (3, 4, 2), (3, 4, 3), (3, 4, 4), (3, 4, 5), (3, 4, 6), (3, 5, 1), (3, 5, 2), (3, 5, 3), (3, 5, 4), (3, 5, 5), (3, 5, 6), (3, 6, 1), (3, 6, 2), (3, 6, 3), (3, 6, 4), (3, 6, 5), (3, 6, 6), (4, 1, 1), (4, 1, 2), (4, 1, 3), (4, 1, 4), (4, 1, 5), (4, 1, 6), (4, 2, 1), (4, 2, 2), (4, 2, 3), (4, 2, 4), (4, 2, 5), (4, 2, 6), (4, 3, 1), (4, 3, 2), (4, 3, 3), (4, 3, 4), (4, 3, 5), (4, 3, 6), (4, 4, 1), (4, 4, 2), (4, 4, 3), (4, 4, 4), (4, 4, 5), (4, 4, 6), (4, 5, 1), (4, 5, 2), (4, 5, 3), (4, 5, 4), (4, 5, 5), (4, 5, 6), (4, 6, 1), (4, 6, 2), (4, 6, 3), (4, 6, 4), (4, 6, 5), (4, 6, 6), (5, 1, 1), (5, 1, 2), (5, 1, 3), (5, 1, 4), (5, 1, 5), (5, 1, 6), (5, 2, 1), (5, 2, 2), (5, 2, 3), (5, 2, 4), (5, 2, 5), (5, 2, 6), (5, 3, 1), (5, 3, 2), (5, 3, 3), (5, 3, 4), (5, 3, 5), (5, 3, 6), (5, 4, 1), (5, 4, 2), (5, 4, 3), (5, 4, 4), (5, 4, 5), (5, 4, 6), (5, 5, 1), (5, 5, 2), (5, 5, 3), (5, 5, 4), (5, 5, 5), (5, 5, 6), (5, 6, 1), (5, 6, 2), (5, 6, 3), (5, 6, 4), (5, 6, 5), (5, 6, 6), (6, 1, 1), (6, 1, 2), (6, 1, 3), (6, 1, 4), (6, 1, 5), (6, 1, 6), (6, 2, 1), (6, 2, 2), (6, 2, 3), (6, 2, 4), (6, 2, 5), (6, 2, 6), (6, 3, 1), (6, 3, 2), (6, 3, 3), (6, 3, 4), (6, 3, 5), (6, 3, 6), (6, 4, 1), (6, 4, 2), (6, 4, 3), (6, 4, 4), (6, 4, 5), (6, 4, 6), (6, 5, 1), (6, 5, 2), (6, 5, 3), (6, 5, 4), (6, 5, 5), (6, 5, 6), (6, 6, 1), (6, 6, 2), (6, 6, 3), (6, 6, 4), (6, 6, 5), (6, 6, 6)]

56 von 216  
 p = 0.25925925925925924

#### Aufgabe 4

Diese Änderung ist insofern nützlich, da einem erspart wird „Augenzahlen“ so oft wiederholt noch einmal schreiben zu müssen wie man es haben möchte. In unserem Fall möchten wir das kartesische Produkt von jeweils drei Augenzahlen erhalten also reicht es aus anstatt product (Augenzahlen, Augenzahlen, Augenzahlen) einfach „Augenzahlen, repeat = 3“ schreiben.