Übungsblatt 4

1. Programm in Python eingeben:

```python
import itertools

def X(ω):
    """Studenten können diese Funktionen selbst programmieren.
    X ist Zufallsvariable, also X: Ω → R.
    Einem Ereignis wird eine reele Zahl zugeordnet."""
    return ω[0]+ω[1]

Augenzahlen = {1, 2, 3, 4, 5, 6}
Ω = list(itertools.product(Augenzahlen, Augenzahlen))
# Ereignis E
E = []
for ω in Ω:
    if X(ω) == 10:
        E.append(ω)
print(f"{len(Ω)} mögliche Ergebnisse")
print (f"{len(E)} günstige Ergebnisse")
print(f"p = {len(E)/len(Ω)}")
```

```
36 mögliche Ergebnisse
3 günstige Ergebnisse
p = 0.08333333333333333
```

Dasselbe Programm wie oben nur ohne Kommentare (da es so übersichtlicher ist):

```python
import itertools
def X(ω):
    return ω[0]+ω[1]
Augenzahlen = {1, 2, 3, 4, 5, 6}
Ω = list(itertools.product(Augenzahlen, Augenzahlen))
E = []
for ω in Ω:
    if X(ω) == 10:
        E.append(ω)
print(f"{len(Ω)} mögliche Ergebnisse")
print (f"{len(E)} günstige Ergebnisse")
print(f"p = {len(E)/len(Ω)}")
```

```
36 mögliche Ergebnisse
3 günstige Ergebnisse
p = 0.08333333333333333
```

a) → *Programm funktioniert richtig*

b) **Unterschied zu Programm aus Blatt 3 (Aufgabe 1):**

→ *Ausgabefeld ist durch unterschiedliche „Print-Angabe" in den letzten Zeilen (beziehungsweise bei Blatt 3 bereits in einer oberen Zeile) unterschiedlich: Es wird im aktuellen Programm nicht mehr der gesamte Ereignisraum sowie günstige von möglichen Ereignissen geschrieben, sondern nun jedes Ereignis (also günstig und möglich) in einer eigenen Zeile ausgegeben. Das Ergebnis für p ist in beiden Programmen gleich.*

→ *Im neuen Programm wird nicht befohlen die beiden Augensummen eines Wertepaares aus Ω zusammenzuzählen, sondern es wird extra eine Funktion definiert, die das automatisch jeweils bei jedem Wertepaar macht (zwar die Funktion X).*

→ *Ebenfalls werden nicht mehr extra die Variablen „möglich“ und „günstig“ definiert, sondern es wird eine Menge E definiert, in die die günstigen Wertepaare (bei denen die Überprüfung in Zeile 8 stimmt) eingefügt werden.*
Erst bei den „Print-Funktionen“ am Schluss wird in dem aktuellen Programm festgelegt, dass für möglichen Ergebnisse in der Ausgabe die Anzahl der Wertepaare in Ω und für günstige Ergebnisse die Anzahl der Wertepaare in E ausgegeben wird.

2. Änderung des Programms, um angegebenes Beispiel lösen zu können:

→ *Es musste zusätzlich Zeile 10 auf 4 Augensummen geändert werden*

```
1  import itertools
2  def X(ω):
3      return ω[0]+ω[1]+ω[2]+ω[3]
4  Augenzahlen = {1, 2, 3, 4, 5, 6}
5  Ω = list(itertools.product(Augenzahlen, Augenzahlen,
       Augenzahlen, Augenzahlen))
6  E = []
7  for ω in Ω:
8      if 7 <= X(ω) <= 14:
9          E.append(ω)
10 print(f"{len(Ω)} mögliche Ergebnisse")
11 print (f"{len(E)} günstige Ergebnisse")
12 print(f"p = {len(E)/len(Ω)}")
13
```

```
1296 mögliche Ergebnisse
706 günstige Ergebnisse
p = 0.5447530864197531
```

→ *Anna wird bei diesem Spiel begünstigt, da die Wahrscheinlichkeit p dafür, dass die Summe der vier addierten Augenzahlen zwischen 7 und 14 liegt, bei zirka 0,545 liegt. Also liegt die Wahrscheinlichkeit, dass eine Augenzahlkombination kommt, bei der Anna gewinnt, bei zirka 54,5 %.*

3. Änderung der Spielregeln in 2, um faires Spiel zu bekommen & Überprüfung durch geeignetes Programm:

1 <u>Änderung der Spielregeln:</u>

→ *Die Regeln müssten so geändert werden, dass Anna nur dann gewinnt, wenn die Summe der vier addierten Augenzahlen zwischen 9 und 14 liegt.*
Es wäre zwar trotzdem noch immer nicht 100% fair, da die Wahrscheinlichkeit für eine Augenzahlenkombination bei der Anna gewinnt, noch immer bei p = 0,50231, also bei zirka 50,23 % liegt.
→ *Anna würde mit diesen Regeln noch mit sechs mehr Augenzahlkombinationen gewinnen können als Bernhard (Anna gewinnt bei 651 Kombinationen und Bernhard bei 645 Kombinationen).*
Es würde auch auf dasselbe Ergebnis für p kommen, wenn man die Zahlen bei denen Anna gewinnt, so setzen würde, dass Anna dann gewinnt, wenn die Augensummen zwischen 14 und 19 liegen würde

Um ein ganz faires Spiel garantieren zu können, müsste der Wert für p bei genau 0,5, also 50 % liegen.
Leider habe ich das trotz langem Probieren nicht geschafft, also war p = 0,50231 der beste Wert, um ein faires Spiel zu garantieren.

2 <u>Programm zum Überprüfen:</u>

```
1  import itertools
2  def X(ω):
3      return ω[0]+ω[1]+ω[2]+ω[3]
4  Augenzahlen = {1, 2, 3, 4, 5, 6}
5  Ω = list(itertools.product(Augenzahlen, Augenzahlen,
       Augenzahlen, Augenzahlen))
6  E = []
7  for ω in Ω:
8      if 9 <= X(ω) <= 14:
9          E.append(ω)
10 print(f"{len(Ω)} mögliche Ergebnisse")
11 print (f"{len(E)} günstige Ergebnisse")
12 print(f"p = {len(E)/len(Ω)}")
13
```

```
1296 mögliche Ergebnisse
651 günstige Ergebnisse
p = 0.5023148148148148
```

4. Zeile 12-15 durch angegebene Zeile ersetzen & Zeile in mathematischer Form als Mengendefinition angeben:

1 <u>Ersetzen der Zeilen 12-15:</u>

```
1  import itertools
2  def X(ω):
3      return ω[0]+ω[1]
4  Augenzahlen = {1, 2, 3, 4, 5, 6}
5  Ω = list(itertools.product(Augenzahlen, Augenzahlen))
6  E = [ω for ω in Ω if X(ω) == 10]
7  print(f"{len(Ω)} mögliche Ergebnisse")
8  print (f"{len(E)} günstige Ergebnisse")
9  print(f"p = {len(E)/len(Ω)}")
10
```

```
36 mögliche Ergebnisse
3 günstige Ergebnisse
p = 0.08333333333333333
```

→ *Wir bekommen dasselbe Resultat, wie im ursprünglichen Programm*

2 <u>Mathematische Form als Mengendefinition:</u>

$$E = \{\omega \mid \omega \in \Omega \wedge X(\omega) = 10\} \quad \text{mit } X(\omega) = \omega_1 + \omega_2$$

E ist die Menge aller Zahlenpaare ω aus der Menge Ω, für die gilt, dass der Wert der Funktion X(ω) gleich 10 ist (mit X(ω) ist die Summer der Augensummen eines Wertepaares).