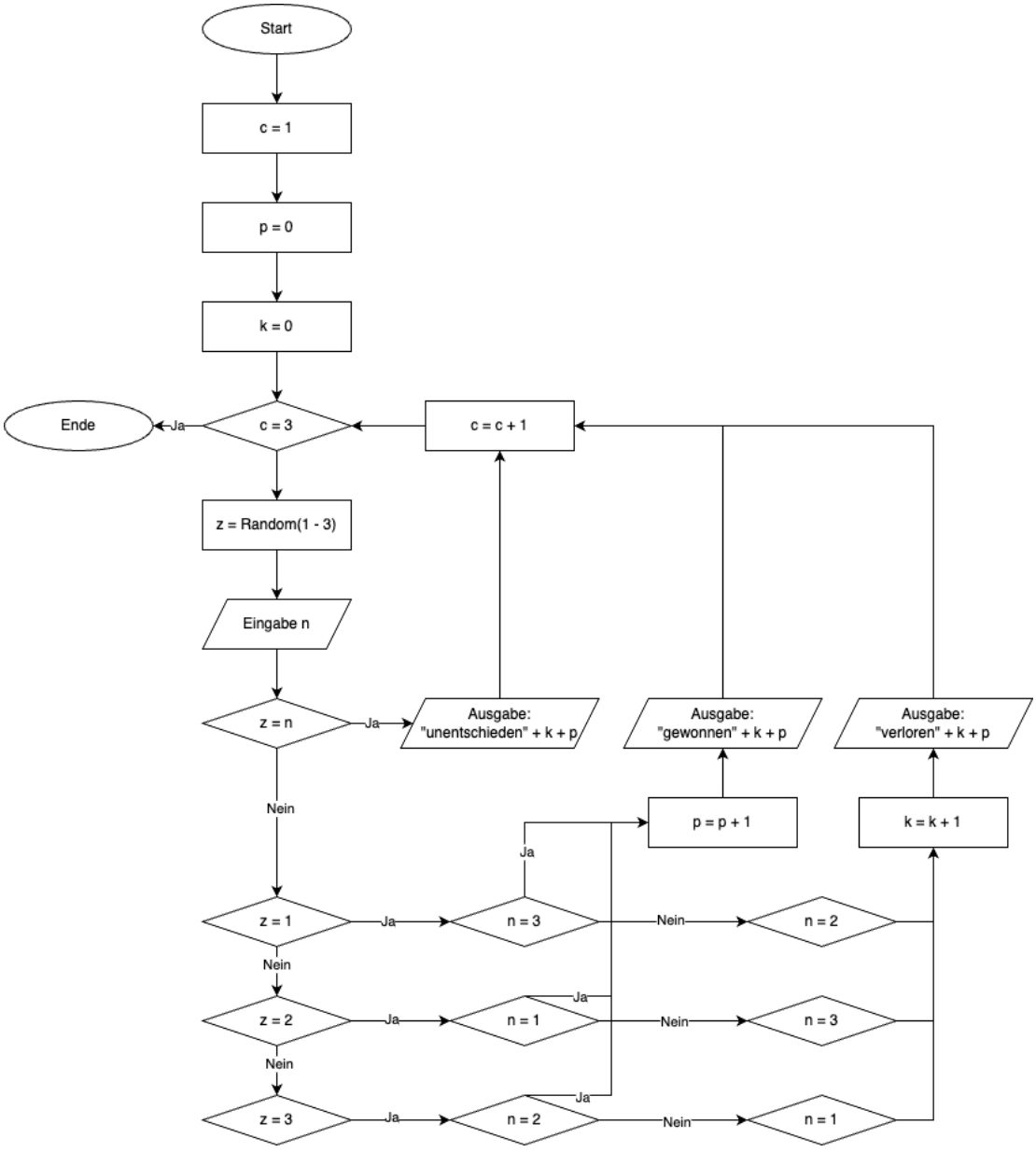
Programmbeschreibung und Flussdiagramm

Hinweis

Dieses Programm spielt mit dem User „Stein, Schere, Papier“. Dabei werden Punkte vergeben.

Die Umsetzung des Programm verwendet:
- while-Schleife
- if-Abfrage
- Variablen
- Inkrementieren



Schritt-für-Schritt-Anleitung

In dem folgenden Teil führt „Schritt-für-Schritt“ durch das Flussdiagramm.

Am Ende dieser Anleitung steht das vollständige Programm passend zum Flussdiagramm.

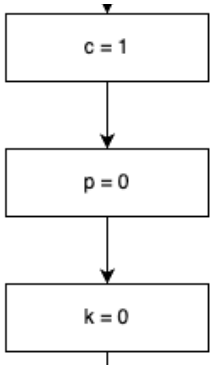
1. Schritt: Die Eingabe (*Zeile 1 und 3*)
Die Variablen z, g und c werden deklariert. **c** dient zum „Hochzählen“, um die Anzahl der Versuche einzuhalten. **p** zählt die Punkte vom User und **k** die Punkte vom „Computer“.

Python
```python
from random import *

def getGewonnen():
  global p
  p += 1
  print("Gewonnen " , str(p) , "/" , str(k))

def getVerloren():
  global k
  k += 1
  print("Verloren " , str(p) , "/" , str(k))

c = 1
p = 0
k = 0
```



2. Schritt: Die while-Schleife (Zeile 5 bis 7)
Solange **c kleiner 4** (also 3 Versuche) ist, läuft das Spiel durch.
Die Variable **n** wird mit dem Wert aus **"prompt**" deklariert.
Der Zähler z erzeugt Zufallszahlen: 1 für Stein, 2 für Schere und 3 für Stein.

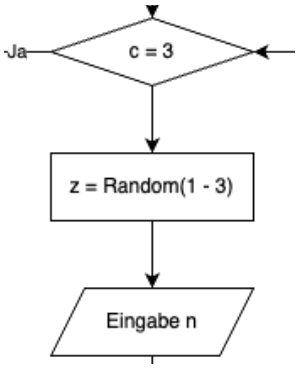
Python
```python
from random import *

def getGewonnen():
  global p
  p += 1
  print("Gewonnen " , str(p) , "/" , str(k))

def getVerloren():
  global k
  k += 1
  print("Verloren " , str(p) , "/" , str(k))

c = 1
p = 0
k = 0

while c < 4:
```



3. Schritt: Die Abzweigungen (Zeile 21 bis 39) und Funktionen (Zeile 3 bis 11)
Die Zeile 1 lädt die Bibliothek, um die Zufallszahlen generieren zu können.

Da sich die Aussagen wiederholen zu „Gewonnen“ und „Verloren“ gibt es zwei Funktionen „getVerloren()“ und „getGewonnen()“. Die Abfrage unterscheidet zunächst, ob ein „Unentschieden“ gegeben ist. Danach erfolgt die Fallunterscheidung zwischen „Gewonnen“ und „Verloren“.

Python

```python
from random import *

def getGewonnen():
  global p
  p += 1
  print("Gewonnen " , str(p) , "/" , str(k))

def getVerloren():
  global k
  k += 1
  print("Verloren " , str(p) , "/" , str(k))

c = 1
p = 0
k = 0

while c < 4:
  z = randint(1, 3)
  n = int(input('Wähle Stein (1), Schere (2), Papier (3): '))

  if z == n:
    print('Unentschieden ' + str(p) + "/" + str(k))

  elif z == 1:
    if n == 2:
      getVerloren()
    else:
      getGewonnen()
  elif z == 2:
    if n == 3:
      getVerloren()
    else:
      getGewonnen()
  elif z == 3:
    if n == 1:
      getVerloren()
    else:
      getGewonnen()
  c += 1
```

Aufgaben

① „Baue“ das Programm mit Hilfe der Schritt-für-Schritt-Anleitung nach. Füge in dem Quellcode passend zu den Schritten Kommentare ein, die den jeweiligen Teil in seiner Funktion beschreiben.

Python

```python
# Ein "Hashtag" ergibt einen Zeilenkommentar

"""Drei Anführungszeichen am Anfang und Ende ergeben
einen Kommentar auf mehreren Zeilen"""
```

② Überlege Dir mögliche Verbesserungen für das Programm. Beispielsweise:

- Könnte der Text der Eingabe und der Ausgabe verbessert werden.
- Lässt sich der Schwierigkeitsgrad erhöhen?
- Mit „logischen Operatoren“ wie && (Und-Verknüpfung) und || (Oder-Verknüpfung) könnte sich der Quellcode vereinfachen lassen, oder?

③ Passe das Flussdiagramm und den Quelltext entsprechend Deiner Überlegungen aus Aufgabe 2 an.