

---

## Übungsblatt 3 zur Vorlesung Grundlagen der Programmierung

---

### Übung 1

#### Schere, Stein, Papier

Wir wollen Schere, Stein, Papier spielen und dafür ein kleines Programm schreiben.

Hier die Beschreibung von Wikipedia<sup>1</sup>:

*„Zwei Spieler wählen je eines der drei möglichen Symbole **Schere**, **Stein** oder **Papier** und zeigen dieses dann auf ein Kommando gleichzeitig mit Hilfe einer ihrer Hände an. Da jedes Symbol gegen ein anderes gewinnen oder verlieren kann, steht immer einer der Spieler als Gewinner fest. Zu **Unentschieden** kommt es, wenn beide Spieler dasselbe Symbol wählen. **Das Spiel wird in diesem Fall wiederholt.**“*

Starten Sie mit der Implementierung wie folgt:

```
# bindet ein Modul ein – bitte Zeile ignorieren
import random

print("Lass uns Schere, Stein, Papier spielen!")

schereString = "Schere"
steinString = "Stein"
papierString = "Papier"

# optionen beinhaltet nun eine Liste mit den drei zulässigen
# Eingabe-Optionen
optionen = (schereString, steinString, papierString)

# random.choice(optionen) beinhaltet einen zufällig gewählten
# Eintrag aus der Liste der Eingabe-Optionen
computerEingabe = random.choice(optionen)

# Lassen Sie den Nutzer seine Wahl eingeben
nutzerEingabe = ...

# Prüfen Sie, ob die Nutzer-Eingabe zulässig ist
if nutzerEingabe not in optionen:
    print("Fehlerhafte Eingabe!")

# ab hier folgt nun Ihr eigener Code
...
```

*Hinweis: der Code kann (und sollte) nicht nur ergänzt, sondern auch verändert werden!*

---

<sup>1</sup> Beschreibung kopiert von: [https://de.wikipedia.org/wiki/Schere,\\_Stein,\\_Papier](https://de.wikipedia.org/wiki/Schere,_Stein,_Papier)

Bitte denken Sie an folgende Punkte:

- 1) Wenn der Nutzer eine fehlerhafte Eingabe macht, so wiederholen Sie die Eingabe bitte so lange, bis eine korrekte Eingabe erfolgt ist
- 2) So lange Computer und Nutzer unentschieden spielen soll das Spiel wiederholt werden
- 3) Beenden Sie das Spiel erst, wenn der Nutzer eine *korrekte* Eingabe gemacht hat **und** er entweder *verloren* oder *gewonnen* hat

### Beispiel-Durchlauf 1:

```
Lass uns Schere, Stein, Papier spielen!
Deine Eingabe (Schere, Stein, Papier): Hallo
Fehlerhafte Eingabe!
Deine Eingabe (Schere, Stein, Papier): Schere
Deine Wahl: Schere; Wahl des Computers: Schere
Unentschieden - neue Runde!
Deine Eingabe (Schere, Stein, Papier): Schere
Deine Wahl: Schere; Wahl des Computers: Stein
Du hast verloren!
```

### Beispiel-Durchlauf 2:

```
Lass uns Schere, Stein, Papier spielen!
Deine Eingabe (Schere, Stein, Papier): Papier
Deine Wahl: Papier; Wahl des Computers: Schere
Du hast verloren!
```

### Beispiel-Durchlauf 3:

```
Lass uns Schere, Stein, Papier spielen!
Deine Eingabe (Schere, Stein, Papier): Stein
Deine Wahl: Stein; Wahl des Computers: Schere
Du hast gewonnen!
```

## Übung 2

### Tannenbäume auf der Kommandozeile

Sie möchten auf der Kommandozeile gerne einen vereinfachten Tannenbaum mit Sternchen malen. Hierfür darf der Nutzer die maximale Breite des Baums angeben. Die minimale Breite beträgt 5. Der Fuß des Baums hat immer die Höhe von einer Zeile und eine Breite von drei Zeichen. Zur Vereinfachung soll die Breite immer eine ungerade Zahl sein.

#### Beispiel-Baum 1: Breite 5

```

  *
 * *
* * *
* * * *
 * * *
```

#### Beispiel-Baum 2: Breite 9

```

    *
   * * *
  * * * *
 * * * * *
* * * * *
 * * * * *
  * * * *
   * * *
```

### Beispiel-Baum 3: Breite 11

```

      *
    * * *
  * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
  * * *

```

### Beispiel-Baum 4: Breite 17

```

      *
    * * *
  * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * * *
  * * *

```

## Übung 3

### Einmaleins

Schreiben Sie ein Programm, welches die Zahlen von 1 bis 100 in einer 10x10 Matrix ausgibt.

*Hinweis: Python unterstützt die Formatierung von Strings bei der Ausgabe. So gibt beispielsweise folgender Befehl den Wert der Variable value vom Typ int rechtsbündig mit vier Stellen aus. Mehr Formatierungsmöglichkeiten finden Sie in der Python Dokumentation<sup>2</sup>*

```
# Formatierung des Werts value rechtsbündig mit vier Stellen
str("{:>4}").format(value)
```

### Erwartete Ausgabe:

```

 1   2   3   4   5   6   7   8   9  10
11  12  13  14  15  16  17  18  19  20
21  22  23  24  25  26  27  28  29  30
31  32  33  34  35  36  37  38  39  40
41  42  43  44  45  46  47  48  49  50
51  52  53  54  55  56  57  58  59  60
61  62  63  64  65  66  67  68  69  70
71  72  73  74  75  76  77  78  79  80
81  82  83  84  85  86  87  88  89  90
91  92  93  94  95  96  97  98  99 100

```

<sup>2</sup> <https://docs.python.org/3/library/string.html?highlight=string%20format#custom-string-formatting>

## Übung 4

### Fingerübungen

Schreiben Sie folgende Programme. In allen Aufgaben ist  $n$  ein vom Benutzer angegebener Wert.

- 1) Geben Sie die Summe aller Zahlen von 1 bis  $n$  aus. Bestimmen Sie den Wert bitte, ohne die Gauß'sche Formel zu verwenden.
- 2) Bestimmen Sie die Summe jeder  $n$ -ten Zahl zwischen 1 bis 100.
- 3) Geben Sie alle Primzahlen von 1 bis  $n$  aus. Bei der Bestimmung müssen Sie sich keine besondere Mühe bzgl. der Laufzeitkomplexität Ihrer Lösung geben.
- 4) Finden und benennen Sie alle natürlichen Zahlen zwischen 1 und 100, die durch  $n$  teilbar sind.  $n$  ist ein vom Benutzer angegebener Wert.

## Übung 5

### Das Heron-Verfahren

In der ersten Vorlesung haben wir das Heron-Verfahren kennengelernt. Implementieren Sie bitte das Heron-Verfahren zur Bestimmung der Wurzel aus einer Zahl  $x$  bei gegebenem Startwert  $g$ . Sowohl  $x$  als auch  $g$  sind Eingabe des Nutzers. Terminieren Sie den Algorithmus, wenn die Differenz  $d$  zwischen der letzten und der aktuellen Näherung  $d < 10^{-5}$  ist, spätestens aber nach 5 Iterationen. Geben Sie in jeder Iteration Ihre Zwischenergebnisse aus. Versuchen Sie eigenständig in der Python-Dokumentation (Link siehe Übung 3) herauszufinden, wie man die Werte für  $g$  in jeder Iteration gut lesbar formatiert ausgibt.

*Hinweis: die Funktion  $\text{abs}(x)$  gibt Ihnen den absoluten Wert von  $x^3$ .*

#### Beispiel-Ausgaben:

```
Bestimme die Wurzel aus: 99
Starte mit folgendem Wert: 2
Iteration 1 - Wert von g: 25.7500
Iteration 2 - Wert von g: 14.7973
Iteration 3 - Wert von g: 10.7439
Iteration 4 - Wert von g: 9.9792
Iteration 5 - Wert von g: 9.9499
Die Wurzel aus 99 beträgt näherungsweise 9.9499
```

```
Bestimme die Wurzel aus: 10
Starte mit folgendem Wert: 3
Iteration 1 - Wert von g: 3.1667
Iteration 2 - Wert von g: 3.1623
Iteration 3 - Wert von g: 3.1623
Die Wurzel aus 10 beträgt näherungsweise 3.1623
```

<sup>3</sup> <https://docs.python.org/3/library/functions.html#abs>