

Schulung Blinkenlights

Komplett ohne irgendwas Blinkendes

Sebastian KÖLL, Jan RÜTHER

August 12, 2020

1 Aufgabe 1: Python zum Fliegen bringen

1.1 Aufgabe 1a

Bringt das folgende Programm zum laufen und testet es

```
import math

def main():

    radius = int(input("Kugelradius in Metern angeben;"))

    oberflaeche = 4 * math.pi * radius**2
    volumen = (4/3) * math.pi * radius**3

    print(f" Die Obefleache der Kugel betraegt = {oberflaeche}
          Quadratmeter")
    print(f" Das Volumen der Kugel betraegt = {volumen} Kubikmeter"
          )

if __name__ == '__main__':
    main()
```

1.2 Aufgabe 1b

Schreibt ein Programm (ähnlich dem in Aufgabe 1a), welches die Höhe h und den Radius r eines Zylinders einliest und dann die gesamte Oberfläche und das Volumen des Zylinders berechnet.

1.3 Aufgabe 1c

Schreibt ein Programm, welches für eine positive ganze Zahl die Quersumme berechnet. Die Quersumme von 1337 beispielsweise ist 14.

2 Aufgabe 2: Dinge mit Listen und Schleifen

Schreibt ein Programm, das eine Folge von Komma-Zahlen einliest. Das Ende der Zahlenfolge wird erkannt durch das erste Zeichen, welches keine Zahl (also z.B. ein Buchstabe) ist. Das Programm soll dann für die eingelesenen Zahlen folgende Größen berechnen:

- a. Die Anzahl der eingelesenen Zahlen,
- b. Die Summe der eingelesenen Zahlen,
- c. Das Maximum der eingelesenen Zahlen,
- d. Das Minimum der eingelesenen Zahlen,
- e. Den Mittelwert der eingelesenen Zahlen

3 Aufgabe 3: Streichhölzer ziehen

In dieser Aufgabe soll ein bekanntes Streichholzspiel realisiert werden. Von einer Anfangsmenge von Streichhölzern nehmen zwei Spieler abwechselnd eins, zwei oder drei Hölzchen weg. Wer das letzte Hölzchen nehmen muß, hat verloren. Das Programm soll so gestaltet sein, daß ein Spieler gegen den Computer spielt.

Der Spieler kann dabei bei jedem seiner Züge nach Gutdünken wahlweise eins, zwei oder drei Hölzchen nehmen. Die Züge des Computers seien zunächst in einer ersten Version so realisiert, daß er zufällig eins, zwei oder drei Hölzchen nimmt. Lediglich wenn nur noch vier Hölzchen oder weniger übrig geblieben sind, spielt der Computer so, daß er gewinnt. Bei vier Hölzchen beispielsweise nimmt er drei, damit der Spieler auf dem letzten Hölzchen sitzen bleibt.

Zufallszahlen kann man in python mit der Funktion **randint** aus der Bibliothek **random** generieren. Mit den Anweisungen:

```
import random

krasse_zahl = random.randint(1,10)
```

wird der Variable `krasse_zahl` eine Zufallszahl zwischen 1 und 10 zugewiesen.

Damit bei jedem Programmdurchlauf immer eine neue Folge von Zufallszahlen berechnet wird, muss der Zufallsgenerator einmal am Anfang des Programms initialisiert werden. Dies geschieht mit der Funktion **random.seed**. Mit könnte man somit beispielsweise Zufallszahlen zwischen 1 und 3 erzeugen. Achte darauf, daß auch der Spieler immer nur eins, zwei oder drei Hölzchen nehmen darf, nicht mehr und nicht weniger.

Wenn das Programm läuft, überlegt euch eine optimale Strategie für den Computer. Der Computer soll also in einer verbesserten Version des Programms von Beginn an stets den optimalen Zug ausführen. Er soll also immer so ziehen, daß er, wenn er die Chance zu gewinnen hat, diese auch nutzt. Die Darstellung auf der Console könnte dann z.B. wie folgt aussehen:

Die aktuelle Anzahl der Hölzchen ist: 13

IIIIIIIIIIIIII

IIIIIIIIIIIIII

IIIIIIIIIIIIII

Gönn dir Hölzchen, Brudi! (1, 2, 3):