

Aufgabe 1. Mache Aufgabe 1 und Aufgabe 2 vom gestrigen Tag fertig. Die restlichen Aufgaben sind nicht so wichtig. Die kannst du später bearbeiten. Dabei könntest du hier und da mal ins Skript schauen um dein Wissen zu vertiefen.

Aufgabe 2. Da wir nun wissen wie man eine beliebige Anzahl von Argumenten übergibt, verändere den Polynomkonstruktor, so dass man ein Polynom vom Grad (kleiner oder gleich) n durch Angabe von $n + 1$ Koeffizienten erstellen kann. Zum Beispiel so:

```
1 p = Polynom(1,4,2) # p = x^2 + 4x^1 + 2x^0
```

Aufgabe 3. Lies im Skript ein wenig mehr über Ausnahmen und erweitere die Polynomklasse um Ausnahmen. Zum Beispiel soll im Konstruktor die Ausnahme `raise TypeError('Falscher Typ. Erwarte int oder float.')` vom Typ `TypeError` mit Wert `'Falscher Typ. Erwarte int oder float.'` ausgelöst werden, falls einer der übergebenen Parameter nicht vom Typ `int` oder `float` ist.

Hierbei machen wir darauf aufmerksam wie man überprüft ob ein Objekt `x` eine Instanz der Klasse `kl` ist. Das geschieht mit `isinstance(x, kl)`. Zum Beispiel liefert `isinstance('hi', str)` den Wert `True` zurück.

Aufgabe 4. Erweitere alle anderen Klassen, die du bisher geschrieben hast um Ausnahmen. Du kannst dich hierbei auf die Ausnahmen `ValueError` und `TypeError` beschränken. Diese sollen immer dann greifen, wenn ein falscher Typ oder falscher Wert benutzt wird.

Aufgabe 5. Gegeben sei eine Datei, in der ganze Zahlen (Integer) stehen. Schreibe ein Programm, welches die Zeilen der Datei einliest und nach dem Wert des letzten Eintrags sortiert. Achtung: Die Zeilen in der Datei können verschieden lang sein.

Tip: Die Funktion `split` für strings ist sehr nützlich.

Aufgabe 6. Lies mit `help(sorted)` die Hilfe zur Funktion `sorted` durch. Nutze `sorted` um eine Sequenz zu erstellen, welche alle Zahlen von 1 bis 100 enthält und diese nach ihrer Teilbarkeit durch 3 sortiert. Also Zahlen, die häufiger durch 3 teilbar sind kommen vor Zahlen, die weniger häufig durch 3 teilbar sind. Diese Aufgabe solltest du in einer Codezeile lösen können.

Schreibe nun eine Funktion mit den Parametern `n` und `k`, die eine Sequenz der Zahlen `0, ..., n` zurückgibt, die nach ihrer Teilbarkeit durch `k` geordnet sind.

Aufgabe 7. Lese dir im Skript durch, wie du eigene Ausnahmen erstellen kannst. Erstelle die beiden Ausnahmen `KeineQuadGleichung` und `KeineReelleLsg` und schreibe eine Funktion, welche drei Parameter `a, b, c` bekommt und die Lösungen der quadratischen Gleichung $ax^2 + bx + c = 0$ zurückgibt. Falls die Parameter keine quadratische Gleichung definieren oder die Gleichung keine reelle Lösung hat, sollen die obigen Ausnahmen ausgelöst werden.

Aufgabe* 8. Schreibe eine Klasse, die sich wie ein mathematisches Pendel verhält¹: Das Pendel soll zu gegebenen Startwerten ϕ , g und l erstellt werden können, wobei ϕ der Ausgangswinkel, g die Erdbeschleunigung und l die Fadenlänge ist.

Nun wollen wir das Pendel in diskreten Zeitabschnitten $t \sim 0$ schwingen lassen. Dazu soll die Klasse über eine Memberfunktion `tick()` verfügen, welche die neue Position des Pendels berechnet (nachdem $t \sim 0$ Zeit vergangen ist).

¹Die Bewegungsgleichung findest du auf https://de.wikipedia.org/wiki/Mathematisches_Pendel