

AUFGABEN SESSION 6:

AUFGABE 1: EINKAUFLISTE SORTIEREN



Öffne die Datei „Einkaufsliste.txt“ in Python und lass dir den Inhalt der Einkaufsliste ausgeben. Zusätzlich sollst du jeden Artikel in einer Liste speichern. Sortiere die Liste anschließend alphabetisch und erstelle eine neue Datei mit dem Namen „Einkaufsliste_sortiert.txt“ in der du die sortierte Liste speicherst. Achte darauf, dass du die neue Datei mit Python erstellst und die Artikel jeweils in einer neuen Zeile zu finden sind.

AUFGABE 2: TASCHENRECHNER



Erstelle für einen Taschenrechner die Funktionen „addieren“, „subtrahieren“, „multiplizieren“ und „dividieren“. Die Funktionen sollen diese vier arithmetischen Operationen jeweils mit zwei Werten durchführen. Der errechnete Wert soll als Rückgabeparameter verwendet werden.

AUFGABE 3: TASCHENRECHNER BELIEBIGE PARAMETERANZAHL



Erweitere die vorherigen Funktionen aus Aufgabe 3, mit der Funktionalität, beliebig viele Parameter in die Funktionen übergeben zu können. Es soll am Ende immer noch der errechnete Wert als Rückgabeparameter verwendet werden. Führe abschließend folgende Berechnung mit den Funktionen durch:

$$((1 + 2 + 3) * 4 * 5) / 6$$

AUFGABE 4: DOMINOSTEINE



Erstelle eine Funktion, in welcher Dominosteine erzeugt werden. Diese sollen bis zu einer durch den Benutzer festzulegenden Grenze erstellt werden. Die Steine können beispielsweise so aussehen:

(1/1), (1/2), (2/2), (1/3), ...

Es dürfen keine Duplikate der Dominosteine entstehen, daher wenn ein Stein mit der Kombination (1/2) besteht, darf ein Stein mit der Kombination (2/1) nicht erzeugt werden. Überlege dir hierzu einen geeigneten Mechanismus. Die fertigen Dominosteine sollen abschließend zurückgegeben werden.

AUFGABE 5: NESTED FUNCTIONS



Erstelle eine äußere Funktion „addition“. Diese beherbergt die innere Funktion „eingabe“. Die Funktion „eingabe“ soll mithilfe der Input-Funktion einen int-Wert anfordern und diesen zurückgeben. Hierbei soll ein geeigneter Mechanismus erstellt werden, welcher verhindert, dass ein ValueError die Funktion abbricht. Sollte eine Fehleingabe durch den Benutzer erfolgen sollte dieser die Eingabe erneut vornehmen können. Es soll jedoch auch die maximale Anzahl an falschen Versuchen festgelegt sein. Die eventuell erfolgreich ermittelte Zahl soll dann in der Funktion „addition“ mit 1 addiert werden und zurückgegeben werden.

AUFGABE 6: FIBONACCI ZAHLEN



Schreibe eine Rekursive Funktion „fibonacci“ mit deren Hilfe Fibonaccizahlen an bestimmten Stellen der Fibonacci-Folge berechnet werden können. Die Zahlen ergeben sich aus der Fibonacci-Folge:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 59, ...

Die Zahlen berechnen sich somit immer aus der Addition der beiden vorhergegangenen Zahlen. Mathematisch Definieren sich die Fibonacci-Zahlen folgendermaßen:

$$F_n = F_{n-1} + F_{n-2}$$

$$F_0 = 0$$

$$F_1 = 1$$

Schreibe hierzu auch die entsprechende iterative Funktion zum Berechnen der Fibonacci-Zahlen. Teste die Laufzeit bei kleinen Werten und großen Werten. Was fällt auf?