

Repetition2

Dr. Günter Kolousek

21. Juli 2015

Lege wiederum ein Verzeichnis an. Nennes es `09_repetition2`! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

1 Wiederholung 2

1. Schreibe eine Funktion `compare(a, b)` enthält, die
 - -1 zurückliefert, wenn a kleiner als b ist,
 - 0 zurückliefert, wenn a gleich b ist und
 - 1 zurückliefert, wenn a größer als b ist.
2. Schreibe eine Funktion `slope(x1, y1, x2, y2)`, die zwei Punkte einer Geraden als Parameter erhält und die Steigung k ermittelt und diese zurückliefert.
3. Schreibe eine Funktion `intercept(x1, y1, x2, y2)`, die wiederum zwei Punkte einer Geraden als Parameter erhält und den Schnittpunkt mit der y-Achse ermittelt und zurückliefert.

Diese Funktion soll die Funktion `slope()` verwenden!

4. Schreibe ein Programm `linear.py`, die den Benutzer nach zwei Punkten im zweidimensionalen Koordinatensystem fragt und danach die Parameter der Geradengleichung mit Hilfe der Funktionen `slope()` und `intercept()` berechnet und ausgibt.

Teste!

Teste auch mit den Punkten P(1,5) und Q(1,2)! Was passiert? Warum? Ändere nun die das Programm entsprechend ab, dass dieses nun richtig funktioniert. Verwende `try` und `except`!

5. Berechnung des Nettogehaltes auf Basis der Lohnsteuerbemessungsgrundlage.

Gehen wir davon aus, dass wir eine Funktion schreiben wollen, die das Nettogehalt berechnet, wenn von dem Bruttogehalt (also das, das mit dem Arbeitgeber vereinbart worden ist) schon die Sozialversicherungsbeträge und sonstige Beträge abgezogen worden sind.

Im Gegensatz zum Beispiel aus der ersten Einheit, in dem von einem fixen Prozentsatz ausgegangen wurde, wird hier jetzt der Steuersatz gemäß der folgenden Tabelle berechnet.

Jahreseinkommen	Steuersatz
bis 3000	10%
bis 80000	22%
bis 100000	32%
bis 150000	42%
größer 150000	50%

Schreibe ein Programm, das den Benutzer nach dem Brutto-Jahreseinkommen fragt und danach gemäß der obigen Tabelle das Netto-Jahreseinkommen ausrechnet und anzeigt.

6. Schreibe eine Funktion `spiral(angle, max_size)`, die eine Spirale mit der Turtle gemäß dem folgenden Algorithmus zeichnet:

Für alle `i`, die sich in dem Bereich von `[0, =max_size=)` befinden:
 bewege Turtle um `i` Einheiten in die aktuelle Richtung
 ändere die Richtung der Turtle um `angle` Grad nach links

Teste im interaktiven Interpreter!

7. Schreibe eine Funktion `multiples_of_seven(a, b)`, die genau dann `True` zurückliefert, wenn beide übergebenen Zahlen Vielfache von 7 sind.

Teste im interaktiven Interpreter!

8. Schreibe eine Funktion `elephants1(weight, color, origin)`, die in Abhängigkeit des Gewichtes (engl. `weight`), der Farbe (engl. `color`) und der Herkunft (engl. `origin`), die Kategorie (engl. `category`) eines Elefanten gemäß der folgenden Tabelle bestimmt und zurückliefert:

category	1	2	3	4	5	6	7	8	9	10
weight	3500	3500	3500	4000	4000	4000	4000	4500	4500	4500
color	grey	brown	white	grey	brown	brown	white	grey	brown	white
origin	Africa	Africa	Africa	Africa	Africa	India	India	India	India	India

Das Gewicht darf nicht willkürlich sein, sondern muss jeweils einem der angegebenen Zahlenwerte entsprechen. Ist das übergebene Gewicht nicht in der Tabelle enthalten oder sonst ein Wert nicht gültig, dann soll -1 zurückgeliefert werden.

Teste im interaktiven Interpreter!

9. Schreibe eine Funktion `elephants2(weight, color, origin)`, die wie im vorhergehenden Beispiel funktioniert, jedoch sind jetzt beliebige Gewichtsangaben erlaubt, wobei die angegebenen Gewichte in der Tabelle jetzt jeweils das Maximalgewicht angeben.
10. Schreibe eine Funktion `bmi(weight, height)`, den Bodymaßindex (BMI) berechnet und zurückliefert. Dieser Wert errechnet sich indem das Gewicht in kg durch das Quadrat der Körpergröße in m dividiert wird. Schreibe diese Funktion in eine Datei `bmi.py`.

Ergänze diese Datei um eine Benutzereingabe der beiden Werte und gib dem Benutzer den BMI bekannt.

Wenn die so berechnete Zahl zwischen 18.5 und 26 liegt, dann hat man Normalgewicht. Wenn sie kleiner als 18.5 / größer als 26 ist, dann hat man Untergewicht / Übergewicht.

Ergänze das Programm weiters sodass dem Benutzer bekanntgegeben wird, ob dieser Untergewicht, Normalgewicht oder Übergewicht hat.

11. Schreibe eine Funktion `different(a, b, c)`, die `True` zurückliefert, wenn die 3 übergebenen Zahlen verschieden sind, ansonsten `False`.