

# Übungsblatt 07

## Aufgabe 1

Wir betrachten gemäß der Vorlesung die Klasse `Geld`.

Erweitern Sie die hier gegebene Klassendefinition um alle notwendigen (überladenen) Methoden, damit das Hauptprogramm funktioniert und folgende Ausgabe liefert:

USD 100.0

EUR 202.0

Der Mietwagen ist nicht teurer

```
In [ ]: # TODO Aufgabe 1
class Geld:
    # für das Klassenattribut
    __wechselkurs = {'USD': 0.84998,
                     'GBP': 1.39480,
                     'EUR': 1.0,
                     'JPY': 0.007168}

    # Konstruktormethode
    def __init__(self, waehrung, betrag):
        self.__waehrung = waehrung
        self.__betrag = float(betrag)

    def getEuro(self):
        return self.__betrag*self.__wechselkurs[self.__waehrung]

    # TODO implementieren Sie hier die überladenen Methoden


# Hauptprogramm
print(Geld('USD', 100))
hotelkosten = Geld('USD', 120)
mietwagen = Geld('EUR', 100)
print(mietwagen + hotelkosten)
if mietwagen > hotelkosten:
    print('Der Mietwagen ist teurer')
else:
    print('Der Mietwagen ist nicht teurer')
```

## Aufgabe 2

Programmieren Sie eine Klasse `Bruch` zum Hantieren mit Brüchen. Sie soll über die ganzzahligen Attribute `zaehler` und `nenner` verfügen und alle Methoden aufweisen, damit nachfolgendes Hauptprogramm lauffähig ist:

Folgende Ausgabe sollte zu sehen sein:

```
3/4
3/4
3/1
b1 kleiner b2
0/1
0.75
15/4
15/4
3/10
3/10
45/27
5/3
5
3
3/5
ungleich
```

(wobei die letzten beiden Zeilen der Ausgabe von Ihren Eingaben für `b2` abhängen)

```
In [1]: # TODO Aufgabe 2)
```

```
In [ ]: # Das Testprogramm
b1 = Bruch(3,4)
b1.ausgabe() # 3/4
print(b1) # 3/4

b2 = Bruch(3)
b2.ausgabe() # 3/1

if b1 < b2 :           # b1 kleiner b2
    print("b1 kleiner b2")
else :
    print("b1 größergleich b2")

b3 = Bruch()
b3.ausgabe() # 0/1

print(b1.berechne_wert()) # 0.75
b4 = b1 + b2
print(b4)
b1 += b2
print(b1)

b5 = Bruch(3,4)
b6 = Bruch(2,5)
b1 = b5 * b6
print(b1)
b5 *= b6
print(b5)

b3.initialisieren(45,27)
b3.ausgabe() # 45/27

b3.kuerzen()
b3.ausgabe() # 5/3

print(b3.gib_zaebler()) # 5
print(b3.gib_nenner()) # 3
```

```
b2.eingabe()
b2.ausgabe()

if b2 == b1 :           # Je nach Eingabe
    print("gleich")
else :
    print("ungleich")
```