

LU02.A07 - Buchhaltung

Erstelle ein Python-Programm, das Buchhaltungsdaten in Form von Transaktionen verwaltet. Verwende dabei ausschließlich **Immutable Values** und **Pure Functions**.

Detaillierte Aufgabenstellung

Ihr seid in einer Buchhaltungsabteilung tätig und sollt ein Programm schreiben, das einfache Finanztransaktionen aufnimmt und den aktuellen Kontostand berechnet.

1. Definiere eine Funktion namens `add_transaction`, die den bisherigen Transaktionsverlauf und eine neue Transaktion als Argumente nimmt. Die Funktion soll einen neuen Transaktionsverlauf zurückgeben, der die neue Transaktion enthält.
2. Definiere eine Funktion namens `calculate_balance`, die den Transaktionsverlauf als Argument nimmt und den aktuellen Kontostand berechnet.
3. Verwende nur Immutable Values wie Tupel und Strings für die Transaktionen und den Verlauf.
4. Achte darauf, dass alle deine Funktionen Pure Functions sind.

```
if __name__ == '__main__':  
    # Dein Code kommt hier hin
```

Beispiel Daten

- Anfangs-Transaktionsverlauf: (('Deposit', 1000), ('Withdrawal', 200))
- Neue Transaktionen: ('Deposit', 500), ('Withdrawal', 100), ('Deposit', 300)

Schrittweises Vorgehen

1. **Immutable Values Definieren:** Verwende Tupel zur Speicherung von Transaktionen. Jedes Tupel enthält die Art der Transaktion ('Deposit' oder 'Withdrawal') und den Betrag.
 - Beispiel: ('Deposit', 100), ('Withdrawal', 50)
2. **Pure Functions Anwenden:**
 - Implementiere die `add_transaction` Funktion. Diese nimmt den aktuellen Transaktionsverlauf und eine neue Transaktion als Argumente und gibt einen neuen Transaktionsverlauf zurück.
 - Pseudocode:
 - `def add_transaction(transactions, new_transaction):`
 - `return` neuer Transaktionsverlauf
 - Implementiere die `calculate_balance` Funktion, die den aktuellen Kontostand aus einem gegebenen Transaktionsverlauf berechnet.
 - Pseudocode:
 - `def calculate_balance(transactions):`
 - `return` aktueller Kontostand

3. **Main Funktion:** Verwende die beiden Funktionen, um einige Transaktionen hinzuzufügen und den Kontostand zu berechnen.
- Füge z.B. 3 Einzahlungen und 2 Abhebungen hinzu und berechne dann den Kontostand.
-

[M323-LU02](#), [M323-A1B](#), [M323-A1I](#)

 © Kevin Maurizi

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu02/aufgaben/buchhaltung>

Last update: **2026/08/18 10:20**

