

## Aufgabe 4 - Unit Tests anwenden

### Ziel

- Sie können Testklassen auf der Basis von pytest erstellen und anwenden.

### Auftrag

- Testen Sie die bekannte Klasse BankAccount.
- Benutzen Sie dazu den unten aufgeführten Code.
- Ersetzen Sie die TODO-Anweisungen sinngemäss mit Ihrem Code.

### Abgabe

Geben Sie Ihren Code via Moodle ab.

---

### Code

[bank\\_account.py](#)

```
class BankAccount:  
    """  
        Diese Klasse stellt ein Bankkonto dar, das korrekt realisiert ist.  
    """  
    def __init__(self, starting_amount):  
        """  
            Konstruktor eröffnet das Konto mit einem Startbetrag  
            :param starting_amount:  
        """  
        self.__saldo = starting_amount  
  
    def get_money(self, value):  
        """  
            Es wird der durch value benannte Betrag vom Bankkonto bezogen.  
            Dabei wird sichergestellt, dass der Wert von saldo nicht  
            negativer wird.  
            :param value: zu beziehender Betrag  
            :return: bezogener Betrag  
        """  
        if value < self.__saldo:  
            self.__saldo -= value  
            return value  
        else:  
            return 0
```

```
@property
def saldo(self):
    """
    Liefert den aktuellen Saldo des Kontos
    :return:
    """
    return self.__saldo
```

[bank\\_account\\_test.py](#)

```
import pytest
from bank_account import BankAccount

def test_get_money_less_than_saldo_returns_expected_money():
    # TODO: arrange - create testee with saldo 400
    # TODO: act      - invoke the get_money methode with 300, an amount
    # less than saldo
    # TODO: assert   - make sure that expected value is 300 returned
    # and...
    # TODO:           that balance is 100 as expected
    assert False

def test_get_money_with_more_than_saldo_returns_nothing():
    # TODO: arrange - create testee with saldo 400
    # TODO: act      - invoke the get_money methode with 600, an amount
    # more than saldo
    # TODO: assert   - make sure that expected value is 0 returned
    # and...
    # TODO:           that balance is still 400 as expected
    assert False
```

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

Permanent link:  
[https://wiki.bzz.ch/modul/m320/learningunits/lu90/aufgaben/lu4-aufgabe\\_4](https://wiki.bzz.ch/modul/m320/learningunits/lu90/aufgaben/lu4-aufgabe_4)

Last update: **2024/03/28 14:07**

