2025/11/27 09:02 Lösung 6 - data hiding

Lösung 6 - data hiding

Das **Bankkonto Nr. 1** liefert bei einem Saldo von 700.0 keinen Betrag bzw. den Wert 0.0, wenn 900.0 abgehoben werden sollen.

Hier ist in der entsprechenden Methode get_money eine Zusicherung realisiert.

```
def get_money(self, value):
"""
Es wird der durch value benannte Betrag vom Bankkonto bezogen.
Dabei wird sichergestellt, dass der Wert von saldo nicht
negative wird.
:param value: gewünschter Bezug vom Konto
:return: Betrag, sofern der Saldo genügend gross ist, sonst 0.
"""
if value < self._saldo:
    self._saldo -= value
    return value
else:
    return 0</pre>
```

Das **Bankkonto Nr. 2** liefert bei einem Saldo vom 700.0 den gewünschten Betrag von 900.0, mit dem Effekt, dass der Saldo dann auf -200.0 steht.

Da das Attribut saldo public deklariert ist, kann es von jeder Stelle aus gelesen und geschrieben werden. Selbst wenn der Betrag grösser als der Saldo ist, kann so ein Betrag bezogen werden. Als

Bank wäre man schnell mal bankrott



Die Verantwortung für die Wertzusicherung wird hier dem Aufrufer übertragen. Vergisst er diese zu implementieren, ist das Chaos perfekt. Dies widerspricht einem fundamentalen Grundsatz der OO-Programmierung.



Es gilt der Grundsatz, dass die Verantwortung immer an den Ort der Zuständigkeit gehört.



From:

https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/modul/m320/learningunits/lu01/loesungen/lu1-aufgabe_

Last update: 2024/03/28 14:07



Last update: 2024/03/28 modul:m320:learningunits:lu01:loesungen:lu1-aufgabe_5 https://wiki.bzz.ch/modul/m320/learningunits/lu01/loesungen/lu1-aufgabe_5 https://wiki.bzz.ch/modul/m320/learningunits

https://wiki.bzz.ch/ Printed on 2025/11/27 09:02