# LU02.L03 - Klassendiagramm "BankAccount" umsetzen

bank_account.py

```
""" Module providing a bank account"""


class BankAccount:
    """
    A bank account.

    Attributes
    ----------
    balance: float
        current balance of the bank account
    overdraft: float
        the maximum overdraft amount
    customer: Customer
        the customer this account belongs to

    Methods
    -------
    booking(self, amount):
        adds the amount to the current balance
    get_money(self, amount):
        withdraws the amount from the balance if possible
    """

    def __init__(self, max_overdraft, customer):
        """Constructs a BankAccount object."""
        self._balance = 0.0
        self._overdraft = max_overdraft
        self._customer = customer

    @property
    def balance(self):
        """
        Gets the current balance.

        :return: The current balance(float)
        """
        return self._balance

    @property
    def overdraft(self):
        """
```

```
        Gets the overdraft amount.

        :return: The overdraft amount(float)
        """
        return self._overdraft

    @property
    def customer(self):
        """
        Gets the customer this bankaccount belongs to.

        :return: The customer object(Customer)
        """
        return self._customer

    def booking(self, amount):
        """
        Adds an amount to the balance.

        :param amount: (float): The amount to add
        """
        self._balance += amount

    def get_money(self, amount):
        """
        Withdraws some money if possible.

        :param amount: (float): The amount to add
        """
        if (self._balance + self._overdraft) > amount:
            self._balance -= amount
            return amount
        return 0.0
```

---

m320-LU02

René Probst, bearbeitet durch Marcel Suter

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

Permanent link:
**https://wiki.bzz.ch/de/modul/m320_2024/learningunits/lu02/loesungen/bankaccount**

Last update: **2026/08/30 14:05**