

Класс BankAccount

Реализуйте класс `BankAccount`, описывающий банковский счет. При создании экземпляра класс должен принимать один аргумент:

- `balance` — баланс счета, по умолчанию имеет значение `0`

Экземпляр класса `BankAccount` должен иметь один атрибут:

- `_balance` — баланс счета

Класс `BankAccount` должен иметь четыре метода экземпляра:

- `get_balance()` — метод, возвращающий актуальный баланс счета
- `deposit()` — метод, принимающий в качестве аргумента число `amount` и увеличивающий баланс счета на `amount`
- `withdraw()` — метод, принимающий в качестве аргумента число `amount` и уменьшающий баланс счета на `amount`. Если `amount` превышает количество средств на балансе счета, должно быть возбуждено исключение `ValueError` с сообщением:

На счете недостаточно средств

- `transfer()` — метод, принимающий в качестве аргументов банковский счет `account` и число `amount`. Метод должен уменьшать баланс текущего счета на `amount` и увеличивать баланс счета `account` на `amount`. Если `amount` превышает количество средств на балансе текущего счета, должно быть возбуждено исключение `ValueError` с сообщением:

На счете недостаточно средств

Примечание 1. Числами будем считать экземпляры классов `int` и `float`.

Примечание 2. Дополнительная проверка данных на корректность не требуется. Гарантируется, что реализованный класс используется только с корректными данными.

Sample Input 1:

```
account = BankAccount()
```

```
print(account.get_balance())
account.deposit(100)
print(account.get_balance())
account.withdraw(50)
print(account.get_balance())
```

Sample Output 1:

```
0
100
50
```

Sample Input 2:

```
account = BankAccount(100)

try:
    account.withdraw(150)
except ValueError as e:
    print(e)
```

Sample Output 2:

```
На счете недостаточно средств
```

Sample Input 3:

```
account1 = BankAccount(100)
account2 = BankAccount(200)

account1.transfer(account2, 50)
print(account1.get_balance())
print(account2.get_balance())
```

Sample Output 3:

```
50
250
```

Sample Input 4:

```
account1 = BankAccount(100)
account2 = BankAccount(200)
```

```
try:
    account1.transfer(account2, 150)
except ValueError as e:
    print(e)
```

Sample Output 4:

На счете недостаточно средств