

Mão na massa: Atributos privados e transferência de valores entre contas

Atualmente, conseguimos mudar o valor dos atributos da nossa classe. Por exemplo, conseguimos mudar o saldo da conta simplesmente atribuindo um novo valor a ele:

```
>>> from conta import Conta
>>> conta = Conta(123, "Nico", 55.5, 1000.0)
Construindo objeto ... <conta.Conta object at 0x7fa29f59a518>
>>> conta.saldo = 10000
>>> conta.extrato()
Saldo de 10000 do titular Nico
```

Mas o saldo da conta só deve ser modificado através dos métodos `deposita` e `saca`. Então, para avisar ao desenvolvedor que ele não deve alterar o valor dos atributos acessando-os diretamente, torne-os **privados**, adicionando dois *underscores* à frente dos atributos, por exemplo:

```
def __init__(self, numero, titular, saldo, limite):
    print("Construindo objeto ... {}".format(self))
    self.__numero = numero
    self.__titular = titular
    self.__saldo = saldo
    self.__limite = limite
```

Não esqueça de também modificar os atributos nos métodos.

Transferindo um valor de uma conta para outra

Para transferir um valor de uma conta para outra, crie o método `transfere`, que recebe como argumento uma referência do próprio objeto, o `valor` a ser transferido, e a conta de `destino`. Esse método sacará o valor da conta atual e o depositará na conta de destino.