



UNIVERSIDADE FEDERAL DO CEARÁ
Departamento de Computação

Prof. Lincoln Souza Rocha

CK0442- Técnicas de Programação para Ciência de Dados - T02

TP02
2024.2

1. Criando um Módulo Calculadora Orientado a Objetos

Descrição: Crie um módulo chamado `calculadora_oo.py` e dentro dele implemente a classe `Calculadora` contendo os seguintes métodos que implementam as quatro operações matemáticas básicas: somar, subtrair, multiplicar e dividir. Esses métodos devem receber dois operandos como parâmetro: `operando_a` e `operando_b`. Em seguida, crie um novo arquivo (`testa_calculadora_oo.py`) e, neste arquivo, importe a classe `Calculadora` do módulo `calculadora_oo` (`from calculadora_oo import Calculadora`). Agora, escreva a função teste e, dentro dela, instancie a classe `Calculadora` e exercite os métodos do objeto instanciado com valores fixos (isto é, valores escritos diretamente no código sem a necessidade de solicitar que o usuário entre com os dados). O resultado de todos os testes devem ser impressos na tela usando a função `print`.

2. Adicionando Memória ao Módulo Calculadora Orientado a Objetos

Descrição: O objetivo dessa tarefa é alterar o módulo `calculadora_oo.py` para incluir “memória” acumulativa aos objetos da classe `Calculadora`. Basicamente, o resultado de toda operação realizada por métodos da classe `Calculadora` deve ser armazenado em um atributo privado interno chamado de acumulador (cujo valor inicial deve ser 0.0). Além disso, você deve alterar a implementação dos quatro métodos da classe `Calculadora` para que, se apenas um parâmetro for informado na chamada de alguma operação, o valor da variável acumulador deverá ser considerada no lugar do valor do `operando_a` e o valor informado como parâmetro deverá ser considerado como o `operando_b`. Altere a função teste de `testa_calculadora_oo.py` para acrescentar casos de teste para todos os métodos usando a nova funcionalidade de “memória”. (**Dica.** o `operando_b` deve ter valor *default* igual a `None`)

3. Implementando a classe Conta

Descrição: Crie um módulo chamado `sisbanco.py` e, dentro dele, implemente a classe `Conta` seguindo a especificação abaixo. (**Dica.** A Nota de Aula 05 descreve a implementação completa da classe `Conta`).

```
class Conta:
    def __init__(self, numero: str):
        pass

    def creditar(self, valor: float) -> None:
        pass

    def debitar(self, valor: float) -> None:
        pass

    def get_numero(self) -> str:
        pass

    def get_saldo(self) -> float:
        pass
```

4. Implementando a classe Banco

Descrição: Altere o módulo `sisbanco.py` e, dentro dele, implemente da classe `Banco` seguindo a especificação abaixo. (**Dica.** A Nota de Aula 05 descreve parte da implementação da classe `Banco`).

```
class Banco:
    def __init__(self):
        pass

    def cadastrar(self, conta: Conta) -> None:
        pass

    def procurar(self, numero: str) -> Conta:
        pass

    def creditar(self, numero: str, valor: float) -> None:
        pass

    def debitar(self, numero: str, valor: float) -> None:
        pass

    def saldo(self, numero: str) -> float:
        pass

    def transferir(self, origem: str, destino: str, valor: float) -> None:
        pass
```

5. Interagindo com o Sistema Bancário

Descrição: Crie o módulo `terminal_atendimento.py` e, dentro dele, importe as classes `Conta` e `Banco` do módulo `sisbanco.py`. Em seguida, crie uma função chamada `terminal` que deve simular um terminal de atendimento bancário, isto é, um daqueles terminais onde funcionários de banco atendem os correntistas. Veja a descrição parcial do módulo `terminal_atendimento.py` abaixo.

```
def terminal():
    sisbanco = Banco()
    while(True):
        print("SisBanco::Bem-Vindo!")
        print(" :: Opcoes:: ")
        print("[0]-CriarConta")
        print("[1]-Creditar")
        print("[2]-Debitar")
        print("[3]-Transferir")
        print("[4]-Saldo")
        print("[5]-Sair")
        opcao = input("Digite:")

        if opcao == 0:
            #solicite o numero da conta a ser criada
            #instancie uma conta com esse numero
            #cadastre a conta no sisbanco

        elif opcao == 1:
            #solicite o numero da conta alvo
            #solicite o valor a ser creditado
            #realize a operacao de credito no sisbanco

        elif opcao == 2:
            #solicite o numero da conta alvo
```

```
        #solicite o valor a ser debitado
        #realize a operacao de debito no sisbanco

elif opcao == 3:
    #solicite o numero da conta origem
    #solicite o numero da conta destino
    #solicite o valor a ser transferido
    #realize a operacao de transferencia no sisbanco

elif opcao == 4:
    #solicite o numero da conta alvo
    #realize a operacao de obtencao de saldo no sisbanco
    #exiba o saldo na tela

elif opcao == 5:
    print("SisBanco::Bye!")
    return

if __name__ == "__main__":
    terminal()
```