

**UNIVERSIDADE FEDERAL DO CEARÁ****Departamento de Computação**

Prof. Lincoln Souza Rocha

CK0442- Técnicas de Programação para Ciência de Dados - T02

TP01
2024.2

1. Criando o Módulo Calculadora

Descrição: Crie um módulo chamado `calculadora.py` que contenha funções para operações matemáticas básicas: soma, subtracao, multiplicacao e divisao. As funções devem receber dois operandos como parâmetro: `operando_a` e `operando_b`. Escreva uma função teste em um novo arquivo (`testa_calculadora.py`) para importar o módulo `calculadora` (na importação, use `calc` como alias do módulo `calculadora`) e testar todas as operações com valores de exemplo fixos (isto é, sem a necessidade de solicitar que o usuário entre com os dados). O resultado de todos os testes devem ser impressos usando a função `print`.

2. Adicionando Memória ao Módulo Calculadora

Descrição: O objetivo dessa tarefa é alterar o módulo `calculadora` para incluir “memória”. Basicamente, o resultado de toda operação realizada pelo módulo `calculadora` deve ser armazenado numa variável interna ao módulo chamada de acumulador (cujo valor inicial deve ser 0.0). Além disso, você deve alterar a implementação das quatro operações da 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.py` para acrescentar casos de teste para todas as operações usando a nova funcionalidade de “memória”. (**Dica.** O acumulador deve ser definido como variável global dentro das funções para evitar problemas de escopo de acesso.)

3. Construindo um Módulo com Funções Auxiliares

Descrição: Crie um módulo chamado `conversores.py` que deve implementar as seguintes funções de conversão: `celsius_para_fahrenheit`, `fahrenheit_para_celsius`, `metro_para_pes`, `pes_para_metro`, `quilometro_para_milhas`, `milhas_para_quilometro` e `dia_para_horas`. Em outro arquivo, que deve receber o nome de `testa_conversores.py`, importe este módulo, crie uma função teste e teste todas as funções de conversão. O resultado dos testes devem ser impressos usando a função `print`. (**Dica:** visite o site <https://convertlive.com/pt/converter> para consultar os parâmetros de conversão).

4. Organizando Pacotes

Descrição: Crie um pacote chamado `matematica` com dois módulos internos: `aritmetica.py` (com operações de soma e subtracao) e `geometria.py`, com funções para calcular a área de um círculo (`area_circulo`) e a área de um retângulo (`area_retangulo`). No arquivo principal, com o nome `testa_matematica.py`, importe o pacote `matematica`, crie uma função teste que exercita todas as funções de ambos os módulos (`aritmetica` e `geometria`).

5. Uso de `__init__.py` em Pacotes

Descrição: No pacote `matematica`, adicione um arquivo `__init__.py` que importe automaticamente as funções dos módulos `aritmetica` e `geometria`. No arquivo principal (`testa_matematica.py`), importe diretamente o pacote `matematica` e use suas funções sem precisar importar os módulos internos explicitamente.

6. Subpacotes e Organização Hierárquica

Descrição: Expandir o pacote `matematica` criando um subpacote `matematica/estatistica` que contenha um módulo `media.py`. O módulo `media` deve implementar uma função para calcular a média simples de uma lista de números (`media_simples`). Importe o módulo em `testa_matematica.py` usando o comando `from matematica.estatistica import media`. Altere a função teste e adicione casos de teste para a função `media_simples`.