

# Missão prática – Nível 4 – Mundo 1

## “Conhecendo novos paradigmas”

//Disciplina: Nivel4: Conhecendo novos paradigmas

//Professor: Jhonatan Alves

//Aluno: Rafael Lima de Medeiros

//Turma: 2024.2

//Data atual: 20/08/2024

### Código Calculadora.py

```
Arquivo Editar Seleção Ver ... trabalho_estacio_nivel4

class Calculadora:
    def __init__(self, valorA, valorB, operacao = '+'):
        self.__valorA = valorA
        self.__valorB = valorB
        self.__operacao = operacao

    @property
    def valorA(self):
        return self.__valorA

    @valorA.setter
    def valorA(self, valor):
        self.__valorA = valor

    @property
    def valorB(self):
        return self.__valorB

    @valorB.setter
    def valorB(self, valor):
        self.__valorB = valor

    @property
    def operacao(self):
        return self.__operacao

    @operacao.setter
    def operacao(self, operacao):
        self.__operacao = operacao

    def validarOperacao(self, operacao):
        operacoes_validas = ['+', '-', '*', '/', '^']
        return operacao in operacoes_validas

    def calcular(self):
        if not self.validarOperacao(self.__operacao):
            print("Operação inválida. Digite uma operação válida.")
            exit()

        if self.__operacao == '+':
            return self.__valorA + self.__valorB
        elif self.__operacao == '-':
            return self.__valorA - self.__valorB
        elif self.__operacao == '*':
            return self.__valorA * self.__valorB
        elif self.__operacao == '/':
            if self.__valorB == 0:
                print("Não é possível realizar divisão por zero.")
                exit()
            return self.__valorA / self.__valorB
        elif self.__operacao == '^':
            return self.__valorA ** self.__valorB

    def mostrarResultado(self):
        resultado = self.calcular()
        print(f"Resultado: {self.__valorA} {self.__operacao} {self.__valorB} = {resultado}")
```

### Código main\_calculadora.py

```
Arquivo Editar Seleção Ver ... trabalho_estacio_nivel4

from Calculadora import Calculadora

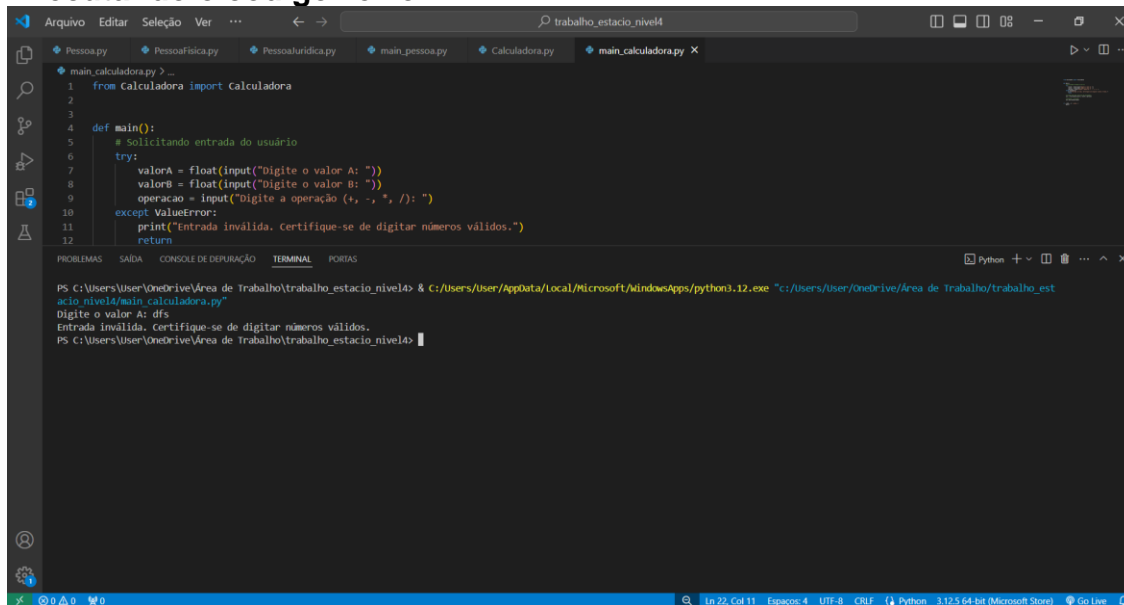
def main():
    # solicitando entrada do usuário
    try:
        valorA = float(input("Digite o valor A: "))
        valorB = float(input("Digite o valor B: "))
        operacao = input("Digite a operação (+, -, *, /, ^): ")
    except ValueError:
        print("Entrada inválida. Certifique-se de digitar números válidos.")
        return

    # Criando uma nova instância da Calculadora
    calc = Calculadora(valorA, valorB, operacao)

    # Exibindo o resultado
    calc.mostrarResultado()

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

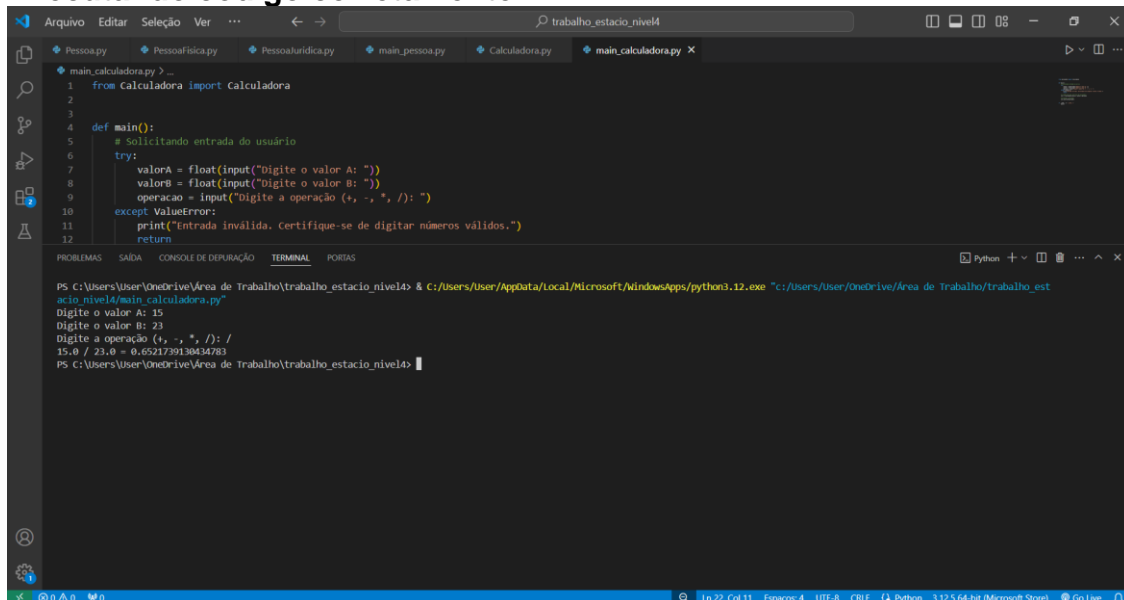
## Executando o código “erro”:



```
1 from Calculadora import Calculadora
2
3
4 def main():
5     # Solicitando entrada do usuário
6     try:
7         valorA = float(input("Digite o valor A: "))
8         valorB = float(input("Digite o valor B: "))
9         operacao = input("Digite a operação (+, -, *, /): ")
10    except ValueError:
11        print("Entrada inválida. Certifique-se de digitar números válidos.")
12    return
```

```
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4> & C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.12.exe "C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4/main_calculadora.py"
Digite o valor A: dfs
Entrada inválida. Certifique-se de digitar números válidos.
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4>
```

## Executando código corretamente:



```
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4> & C:\Users\User\AppData\Local\Microsoft\WindowsApps\python3.12.exe "C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4/main_calculadora.py"
Digite o valor A: 15
Digite o valor B: 23
Digite a operação (+, -, *, /): /
15.0 / 23.0 = 0.6521739130434783
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel4>
```

## Código digitado:

### Calculadora.py

class Calculadora:

```
def __init__(self, valorA=0, valorB=0, operacao='+'):
    self.__valorA = valorA
    self.__valorB = valorB
    self.__operacao = operacao
```

```
@property
def valorA(self):
    return self.__valorA
```

```
@valorA.setter
def valorA(self, valor):
    self.__valorA = valor
```

```
@property
def valorB(self):
    return self.__valorB
```

```
@valorB.setter
```

```

def valorB(self, valor):
    self.__valorB = valor

@property
def operacao(self):
    return self.__operacao

@operacao.setter
def operacao(self, operacao):
    self.__operacao = operacao

def validarOperacao(self, operacao):
    operacoes_validas = ['+', '-', '*', '/']
    return operacao in operacoes_validas

def calcular(self):
    if not self.validarOperacao(self.__operacao):
        print(f"Operação inválida: {self.__operacao}")
        exit()

    if self.__operacao == '+':
        return self.__valorA + self.__valorB
    elif self.__operacao == '-':
        return self.__valorA - self.__valorB
    elif self.__operacao == '*':
        return self.__valorA * self.__valorB
    elif self.__operacao == '/':
        if self.__valorB == 0:
            print("Não é possível realizar divisão por zero.")
            exit()
        return self.__valorA / self.__valorB

def mostrarResultado(self):
    resultado = self.calcular()
    print(f"{self.__valorA} {self.__operacao} {self.__valorB} = {resultado}")

```

## main\_calculadora.py

```
from Calculadora import Calculadora
```

```

def main():
    # Solicitando entrada do usuário
    try:
        valorA = float(input("Digite o valor A: "))
        valorB = float(input("Digite o valor B: "))
        operacao = input("Digite a operação (+, -, *, /): ")
    except ValueError:
        print("Entrada inválida. Certifique-se de digitar números válidos.")
        return

    # Criando uma nova instância da Calculadora
    calc = Calculadora(valorA, valorB, operacao)

    # Exibindo o resultado
    calc.mostrarResultado()

if __name__ == "__main__":
    main()

```