

Relatório de Código: Calculadora em Python

Código:

```
saida=""

def soma(a,b):
    return a+b

def subtracao(a,b):
    return a-b

def multiplica(a,b):
    return a*b

def divisao(a,b):
    if a==0 or b==0:
        print("Não foi possível realizar a divisão por 0")
    else:
        return a/b

def calculadora(a,b,simbolo):
    if simbolo=="+":
        resultado=soma(a,b)
    elif simbolo=="-":
        resultado=subtracao(a,b)
    elif simbolo=="*":
        resultado=multiplica(a,b)
    elif simbolo=="/":
        resultado=divisao(a,b)
    return resultado

while saida!="n":
    a=int(input("Entre com o primeiro valor: "))
    b=int(input("Entre com o segundo valor: "))
    simbolo=str(input("Entre com o simbolo da operação + | - | * | / : "))
    resultado=calculadora(a,b,simbolo)
    print(f"Resultado da operação: {resultado}")
    saida=input("Deseja continuar? S|N :").lower()
```

```
print("Programa encerrado.")
```

Exemplos de Uso:

Exemplos de uso da calculadora:

1. Soma

Entrada: a = 10, b = 5, símbolo = '+'

Saída esperada: 15

2. Subtração

Entrada: a = 10, b = 5, símbolo = '-'

Saída esperada: 5

3. Multiplicação

Entrada: a = 10, b = 5, símbolo = '*'

Saída esperada: 50

4. Divisão

Entrada: a = 10, b = 5, símbolo = '/'

Saída esperada: 2.0

5. Divisão por zero

Entrada: a = 10, b = 0, símbolo = '/'

Saída esperada: "Não foi possível realizar a divisão por 0"

6. Finalizando o programa

Entrada: 'n' no prompt final

Saída: "Programa encerrado."

Comentários Adicionais:

Comentários adicionais:

1. Tratamento de erros: A função 'divisao' impede a divisão por zero, o que é uma prática recomendada.
2. Melhorias sugeridas: Poderíamos modificar a função 'calculadora' para incluir um tratamento adicional quando o símbolo é inválido.
3. Usabilidade: Tornar a entrada de dados mais amigável ao usuário, talvez adicionando uma verificação de entradas válidas para evitar erros.
4. Organização: O uso de funções separadas para cada operação facilita a leitura e a manutenção do código.