

Celio Ludwig Slomp

Soma

Método: `adicao()`

Valores	Entradas de Teste	Saídas
Positivo e Positivo	<code>n1 = 5, n2 = 2</code>	7
Positivo e Negativo	<code>n1 = 5, n2 = -2</code>	3
Negativo e Negativo	<code>n1 = -5, n2 = -2</code>	-7
Valores Inválidos	<code>n1 = None, n2 = 2</code>	<code>RaiseError = TypeError</code>

Subtração

Método: `subtracao()`

Valores	Entradas de Teste	Saídas
Positivo e Positivo	<code>n1 = 5, n2 = 2</code>	3
Positivo e Negativo	<code>n1 = 5, n2 = -2</code>	7
Negativo e Negativo	<code>n1 = -5, n2 = -2</code>	-3
Valores Inválidos	<code>n1 = None, n2 = 2</code>	<code>RaiseError = TypeError</code>

Multiplicação

Método: `multiplicacao()`

Valores	Entradas de Teste	Saídas
Positivo e Positivo	<code>n1 = 5, n2 = 2</code>	10
Positivo e Negativo	<code>n1 = 5, n2 = -2</code>	-10
Negativo e Negativo	<code>n1 = -5, n2 = -2</code>	10
Valores Inválidos	<code>n1 = None, n2 = 2</code>	<code>RaiseError = TypeError</code>

Divisão

Método: divisao()

Valores	Entradas de Teste	Saídas
Positivo e Positivo	n1 = 5, n2 = 2	2.5
Positivo e Negativo	n1 = 5, n2 = -2	-2.5
Negativo e Negativo	n1 = -5, n2 = -2	2.5
Valores Inválidos	n1 = None, n2 = 2	RaiseError = TypeError
Tratando 0	n1 = 5, n2 = 0	RaiseError = ZeroDivisionError

Exponenciação

Método: exponenciacao()

Valores	Entradas de Teste	Saídas
Positivo e Positivo	n1 = 5, n2 = 2	25
Positivo e Negativo	n1 = 5, n2 = -2	0.04
Negativo e Negativo	n1 = -5, n2 = -2	0.04
Valores Inválidos	n1 = None, n2 = 2	RaiseError = TypeError
Tratando Raízes	n1 = -5, n2 = 1/3	RaiseError = ValueError

Raiz Quadrada

Método: raizQuadrada()

Valores	Entradas de Teste	Saídas
Positivo	n1 = 5	2.23606797749979
Negativo	n1 = -5	RaiseError = ValueError
Valores Inválidos	n1 = None	RaiseError = TypeError