

Funcionalidade: Testar o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com argumentos válidos

Cenário: Calcular juros simples

Dado que tenho um capital de 1000, uma taxa de 20% e um período de 10 meses

Quando eu calcular o juros simples com o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 500

Funcionalidade: Testar o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com argumentos inválidos

Cenário: Calcular juros simples com valores string não numéricos

Dado que tenho um capital de 'a', uma taxa de 'a' e um período de 'a' meses

Quando eu calcular o juros simples com o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores inválidos'

Cenário: Calcular juros simples com valores negativos

Dado que tenho um capital de -1000, uma taxa de -10% e um período de -10 meses

Quando eu calcular o juros simples com o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores não podem ser iguais ou menores que zero'

Cenário: Calcular juros simples com valores iguais a zero

Dado que tenho um capital de 0, uma taxa de 0% e um período de 0 meses

Quando eu calcular o juros simples com o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores não podem ser iguais ou menores que zero'

Cenário: Calcular juros simples com tempo sendo valor decimal

Dado que tenho um capital de 1000, uma taxa de 10% e um período de 10.5 meses

Quando eu calcular o juros simples com o método `calcularJurosSimples($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Tempo deve ser um número inteiro'

Funcionalidade: Testar o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com argumentos válidos

Cenário: Calcular juros compostos

Dado que tenho um capital de 1000, uma taxa de 10% e um período de 10 meses

Quando eu calcular o juros compostos com o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 1610.51

Funcionalidade: Testar o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com argumentos inválidos

Cenário: Calcular juros compostos com valores string não numéricos

Dado que tenho um capital de 'a', uma taxa de 'a' e um período de 'a' meses

Quando eu calcular o juros compostos com o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores inválidos'

Cenário: Calcular juros compostos com valores negativos

Dado que tenho um capital de -1000, uma taxa de -10% e um período de -10 meses

Quando eu calcular o juros compostos com o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores não podem ser iguais ou menores que zero'

Cenário: Calcular juros compostos com valores iguais a zero

Dado que tenho um capital de 0, uma taxa de 0% e um período de 0 meses

Quando eu calcular o juros compostos com o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores não podem ser iguais ou menores que zero'

Cenário: Calcular juros compostos com tempo sendo valor decimal

Dado que tenho um capital de 1000, uma taxa de 10% e um período de 10.5 meses

Quando eu calcular o juros compostos com o método `calcularJurosCompostos($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Tempo deve ser um número inteiro'

Funcionalidade: Testar o método `calcularAmortizacao($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com o tipo SAC e argumentos válidos

Cenário: Calcular amortização com o tipo SAC

Dado que tenho um capital de 1000, uma taxa de 10%, um período de 10 meses e o tipo SAC

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser [

'Parcela de Amortização 1' => 100,
'Juros Mês 1' => 100,
'Parcela de Amortização 2' => 100,
'Juros Mês 2' => 90,
'Parcela de Amortização 3' => 100,
'Juros Mês 3' => 80,
'Parcela de Amortização 4' => 100,
'Juros Mês 4' => 70,
'Parcela de Amortização 5' => 100,
'Juros Mês 5' => 60,
'Parcela de Amortização 6' => 100,
'Juros Mês 6' => 50,
'Parcela de Amortização 7' => 100,
'Juros Mês 7' => 40,
'Parcela de Amortização 8' => 100,
'Juros Mês 8' => 30,
'Parcela de Amortização 9' => 100,
'Juros Mês 9' => 20,
'Parcela de Amortização 10' => 100,
'Juros Mês 10' => 10,
'Juros Total' => 550

]

Funcionalidade: Testar o método `calcularAmortizacao($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com o tipo `Price` e argumentos válidos

Cenário: Calcular amortização com o tipo `PRICE`

Dado que tenho um capital de 1000, uma taxa de 10%, um período de 10 meses e o tipo `Price`

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser [

```
'Parcela de Amortização 1' => 62.75,  
'Juros Mês 1' => 100,  
'Parcela de Amortização 2' => 69.02,  
'Juros Mês 2' => 93.73,  
'Parcela de Amortização 3' => 75.92,  
'Juros Mês 3' => 86.82,  
'Parcela de Amortização 4' => 83.51,  
'Juros Mês 4' => 79.23,  
'Parcela de Amortização 5' => 91.87,  
'Juros Mês 5' => 70.88,  
'Parcela de Amortização 6' => 101.05,  
'Juros Mês 6' => 61.69,  
'Parcela de Amortização 7' => 111.16,  
'Juros Mês 7' => 51.59,  
'Parcela de Amortização 8' => 122.27,  
'Juros Mês 8' => 40.47,  
'Parcela de Amortização 9' => 134.5,  
'Juros Mês 9' => 28.25,  
'Parcela de Amortização 10' => 147.95,  
'Juros Mês 10' => 14.8,  
'Juros Total' => 627.45
```

]

Funcionalidade: Testar o método `calcularAmortizacao($capital, $taxa, $tempo)` da classe `CalculadoraFinanceira` com argumentos inválidos

Cenário: Calcular amortização com um tipo diferente de `SAC` e `PRICE`

Dado que tenho um capital de 1000, uma taxa de 10%, um período de 10 meses e o tipo `'teste'`

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser `'Tipo de Amortização Inválido'`

Cenário: Calcular amortização com valores string não numéricos

Dado que tenho um capital de `'a'`, uma taxa de `'a'`, um período de `'a'` e o tipo `Price`

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser `'Valores Inválidos'`

Cenário: Calcular amortização com valores negativos

Dado que tenho um capital de -1000, uma taxa de -10%, um período de -10 meses e o tipo `SAC`

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores não podem ser iguais ou menores que zero'

Cenário: Calcular amortização com valores iguais a zero

Dado que tenho um capital de 0, uma taxa de 0%, um período de 0 meses e o tipo Price

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Valores não podem ser iguais ou menores que zero'

Cenário: Calcular amortização com tempo sendo valor decimal

Dado que tenho um capital de 1000, uma taxa de 10%, um período de 10.5 meses e o tipo Price

Quando eu calcular a amortização com o método `calcularAmortizacao($capital, $taxa, $tempo, $tipo)` da classe `CalculadoraFinanceira`

Então o resultado deve ser 'Tempo deve ser um número inteiro'