

Programação funcional

Lista 1 - Conceitos introdutórios

A lista a seguir contém exercícios introdutórios em Haskell, focados em funções, operadores básicos, condicionais e composição de expressões.

1. Implemente uma função *meuAbs* que devolve o valor absoluto sem usar *abs*.

Exemplo:

meuAbs (-5) → 5

meuAbs 7 → 7

2. Crie uma função que receba dois números e devolva o maior deles.

Exemplo:

maior2 10 7 → 10

maior2 (-3) 5 → 5

3. Escreva uma função que diga se um número é positivo, negativo ou zero.

Exemplo:

classificaSinal (-8) → "Negativo"

classificaSinal 0 → "Zero"

classificaSinal 42 → "Positivo"

4. Crie uma função que classifique uma nota (0 a 10) em conceitos: Excelente, Bom, Regular, Insuficiente.

Exemplo:

conceito 9.5 → "Excelente"

conceito 7.0 → "Bom"

conceito 5.5 → "Regular"

conceito 3.0 → "Insuficiente"

5. Escreva funções que retornem o maior e o menor de três números.

Exemplo:

maior3 4 9 2 → 9

menor3 4 9 2 → 2

6. Crie uma calculadora simples que receba dois números e uma operação ('+', '-', '*', '/').

Exemplo:

calc 8 3 '+' → 11

calc 8 3 '-' → 5

calc 8 3 '' → 24*

calc 8 2 '/' → 4

7. Faça uma função que classifique um triângulo como Equilátero, Isósceles ou Escaleno.

Exemplo:

tipoTriangulo 3 3 3 → "Equilátero"

tipoTriangulo 3 3 2 → "Isósceles"

tipoTriangulo 3 4 5 → "Escaleno"

8. Crie uma função que calcule a diferença absoluta entre dois números.

Exemplo:

difAbs 10 3 → 7

difAbs 3 10 → 7

9. Escreva uma função que diga se um número é múltiplo de outro.

Exemplo:

multiploDe 15 5 → True

multiploDe 14 5 → False

10. Defina a função *min3* que, dados 3 inteiros, retorna o menor deles.

Exemplo:

min3 7 2 9 → 2

min3 (-1) 0 (-5) → -5

11. Defina a função *dezenas* que dado um inteiro toma o valor do dígito das dezenas.

Ex: dezenas 1234 é 3

12. Defina a função *centenas* que dado um inteiro toma o valor do dígito das centenas.

ex: centenas 1234 é 2