

Lambdas

1 - Crie 4 funções lambdas para as operações de matemática básica: soma, subtração, multiplicação e divisão. Cada lambda receberá 2 parâmetros para realizar as operações.

Crie uma função Main e execute-as, printando os resultados.

2 - Crie uma função lambda que receba um e-mail e valide-o para saber se está conforme as regras básicas. Para cumprir a regra básica principal, o mesmo deve conter um @. OBS. essa função deve retornar true ou false, ok? =)

Execute-o na função Main, printando o resultado.

3 - Crie uma função lambda que receba uma `List<Int>` e calcule e retorne o produto dos itens contidos na mesma.

Execute-o na função Main, printando o resultado.



```

//Exemplo simples, retornando UNIT
val soma = { a: Int, b: Int -> println(a + b) }

//Exemplo simples retornando valor, não mais UNIT
val somaComRetorno: (Int, Int) -> Int =
{a: Int, b: Int -> a + b}

//Exemplo mais detalhado em relacao aos parametros e retorno
UNIT

val somaDeOutraFormaEPrinta: (Int, Int) -> Unit =
{ a: Int, b: Int -> println(a + b) }

//Exemplo mais detalhado em relacao aos parametros porém
simplificado após chave e retorno UNIT

val somaDeOutraFormaEPrintaSimplificado: (Int, Int) -> Unit =
{ a, b -> println(a + b) }

//Exemplo com retorno de String

val somaDeOutraFormaERetornaString: (Int, Int) -> String =
{ a: Int, b: Int ->
    val resultado = a + b
    "$resultado"
}

//Exemplo com um unico parametro, para usar o IT

val somaComUmUnicoParametro: (Int) -> Int = {
    it + 10
}

//Exemplo recebendo list como parametro e printnado apenas os
números pares

val listaFiltrada: (List<Int>) -> Unit = {
    for(item in it){
        if(item % 2 == 0){
            println(item)
        }
    }
}

```

