



Lambda Expressions



...

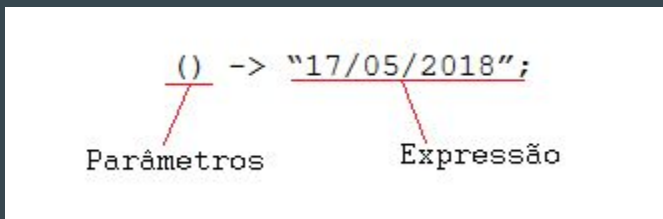
Fernando Pontelo, Jonathas Leandro, Mateus Ferreira e Ravi Assis

O que é uma Expressão Lambda e Interfaces Funcionais?

Expressão lambda é, basicamente, um método anônimo que implementa uma interface funcional. Se assemelham muito com as antigas classes anônimas, na tarefa de implementar interfaces funcionais, mas são mais concisas e mais fáceis de serem utilizadas. Um exemplo é que expressões lambdas implementam interfaces funcionais para serem utilizadas posteriormente. Interfaces funcionais são interfaces que contém apenas um método abstrato e que especifica a finalidade pretendida para a interface. Elas costumam representar uma única ação. Exemplo: Interface Runnable é uma interface funcional porque só define um método: `run()`.

Sintaxe

As expressões lambda introduzem o elemento de sintaxe ‘->’, conhecido como operador lambda ou operador seta. Ele divide a expressão lambda da seguinte forma: do lado esquerdo ao operador lambda (->) é especificado os parâmetros da expressão. Do lado direito é especificado as ações da expressão. Exemplo básico de uma expressão lambda:



Utilizando Expressões Lambda

```
public static void main (String args[]) {  
    TestNum igual = (m, n) -> m == n;  
    if( igual.test(5, 5) ) System.out.println("Numeros iguais");  
    if( !igual.test(10, 2) ) System.out.println("Numeros diferentes");  
  
    TestNum menor = (m, n) -> m < n;  
    if( menor.test(10, 5) ) System.out.println("10 é menor que 5");  
    if( !menor.test(5, 10) ) System.out.println("5 não é menor que 10");  
  
    TestNum fator = (n, d) -> n % d == 0;  
    if( fator.test(10, 2) ) System.out.println("10 é múltiplo de 2");  
    if( !fator.test(10, 3) ) System.out.println("10 não é múltiplo de 3");  
}
```

Referência de Métodos

Uma referência de método é uma forma de referenciar um método sem executá-lo. Como as expressões lambdas, a referência de método precisa de uma interface funcional compatível. Para a referência de métodos é definido um novo operador, o operador ‘ :: ’. Exemplo de sintaxe básica:

```
NomeClasse::nomeMetodo
```

O exemplo acima é de uma referência de um método estático. As referências de métodos também podem ser de métodos de instâncias ou de um construtor de uma classe.

Streams

Uma Stream representa uma sequência de elementos, ela dá suporte a várias operações em sequência nesses elementos. Essas sequências de operações são chamadas de pipeline. Segue um exemplo:

```
List<Integer> numeros = Arrays.asList(1, 2, 3, 4, 5);
```

```
numeros.stream()  
    .filter( n -> n > 2)  
    .forEach( System.out::println );
```