

Lambdas Expression

Por Bruna Dalmagro

Lambdas são funções anônimas, ou seja, funções sem um nome fixo, que podem ser tratadas como dados em linguagens de programação.

São usados para representar blocos de código que podem ser passados como argumentos para funções, armazenados em variáveis e usados de maneira concisa e flexível em programação funcional e OPP com eventos.

No exemplo temos apenas um método para encontrar a *string referida*, porém quando temos apenas 1 método abstrato chamamos de **Functional Interface** e podemos colocar uma anotação indicando, isso é uma boa prática pois assim o compilador irá garantir que você tenha exatamente um método abstrato em sua interface



```
Printable.java

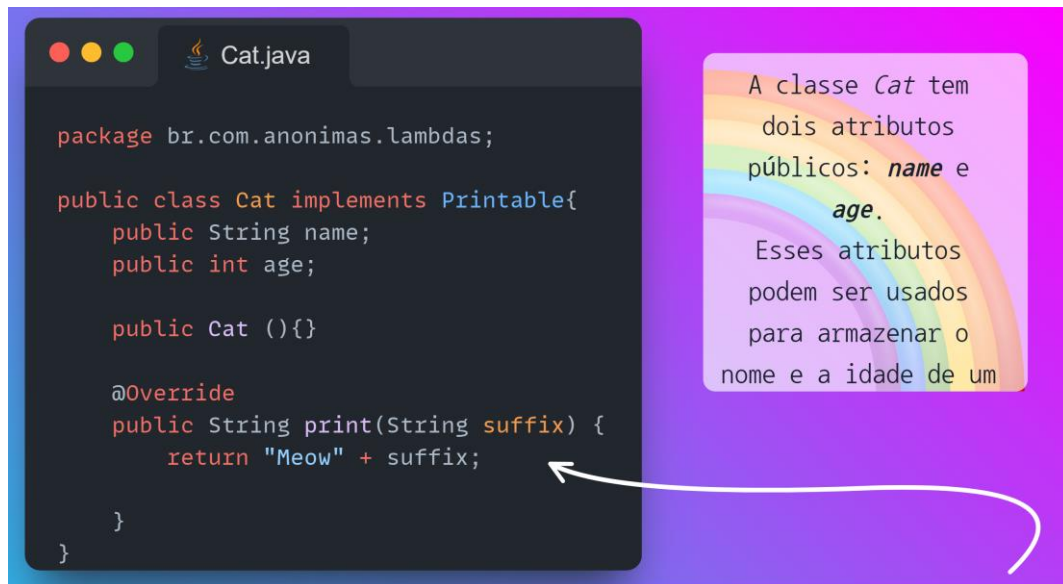
package br.com.anonimas.lambdas;

@FunctionalInterface
public interface Printable {
    String print(String suffix);
}
```

As vezes também chamados de SAM Interfaces (Single Abstracts Methods)

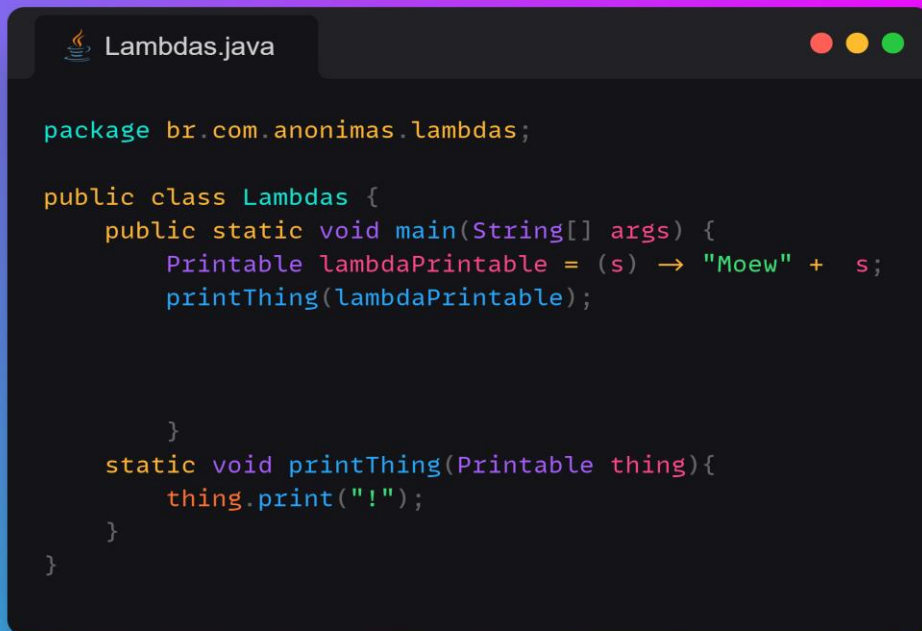
A classe *Cat* fornece uma implementação do método *print* da interface **Printable**. Neste caso, o método *print* concatena a string **"Meow"** com o sufixo passado como argumento (**suffix**) e retorna o resultado.

A instância **printThing** chama o método **print("!!")** da classe **Cat** e nessa instância, ele retornará a string **"Meow!!"**.



Então ao invés de ter que criar uma classe que implementa a interface funcional (***Printable***) e em seguida adicionar a implementação desejada desse único método abstrato e, em seguida, criar um objeto desta classe e usá-lo, você pode simplesmente criar uma expressão ***lambda*** que contém a implementação daquele método de interface funcional que você deseja e que permite que você trate efetivamente o código como um parâmetro.

Como no exemplo a seguir ->



```
package br.com.anonimas.lambdas;

public class Lambdas {
    public static void main(String[] args) {
        Printable lambdaPrintable = (s) -> "Moew" + s;
        printThing(lambdaPrintable);
    }

    static void printThing(Printable thing){
        thing.print("!");
    }
}
```

Printable lambdaPrintable = (s) -> "Moew" + s; cria uma instância de Printable.

A expressão ***lambda (s) -> "Moew" + s*** define a implementação do método print da interface Printable. Neste caso, a expressão ***lambda*** recebe um argumento ***s*** e retorna a concatenação da string ***"Moew"*** com o sufixo ***s***.

Conclusão

De uma perspectiva técnica um *lambda* é um atalho para definir uma implementação de uma interface funcional.