09

Mãos na massa: Lambdas

1) Ainda na classe `Teste` vamos diminuir a burocracia de nosso código. Começaremos usar uma lambda expression para substituir toda classe anônima. Altere o método `sort` :

```
lista.sort( (c1, c2) -> Integer.compare(c1.getNumero(), c2.getNumero()) );
```

2) Analogamente para o trecho de código:

```
Comparator<Conta> comp = new Comparator<Conta> {

    @Override
    public int compare(Conta c1, Conta c2) {
        String nomeC1 = c1.getTitular().getNome();
        String nomeC2 = c2.getTitular().getNome();
        return nomeC1.compareTo(nomeC2);
    }
};
```

Use o seguinte:

```
Comparator<Conta> comp = (Conta c1, Conta c2) -> {
                String nomeC1 = c1.getTitular().getNome();
                String nomeC2 = c2.getTitular().getNome();
                return nomeC1.compareTo(nomeC2);
        };
```

3) Também podemos fazer a mesma coisa para o nosso `System.out.println` , deixando-o bem mais elegante e legível.

Faço o laço de modo seguinte:

```
lista.forEach( (conta) -> System.out.println(conta + ", " + conta.getTitular().getNome()));
```

Uma vez acostumado com a sintaxe dos lambda escrevemos menos códigos sem prejudicar a legibilidade.