10

Mão na massa: Ordenar pelo String

1) Agora começaremos a implementar o método de ordenação baseado no nome do titular da conta.

Para tal crie uma nova classe de teste (ou usa a antigo) que possui o bloco de código abaixo:

```
Conta cc1 = new ContaCorrente(22, 33);
Cliente clienteCC1 = new Cliente();
clienteCC1.setNome("Nico");
cc1.setTitular(clienteCC1);
cc1.deposita(333.0);

Conta cc2 = new ContaPoupanca(22, 44);
Cliente clienteCC2 = new Cliente();
clienteCC2.setNome("Guilherme");
cc2.setTitular(clienteCC2);
cc2.deposita(444.0);

Conta cc3 = new ContaCorrente(22, 11);
Cliente clienteCC3 = new Cliente();
clienteCC3.setNome("Paulo");
cc3.setTitular(clienteCC3);
cc3.deposita(111.0);

Conta cc4 = new ContaPoupanca(22, 22);
Cliente clienteCC4 = new Cliente();
clienteCC4.setNome("Ana");
cc4.setTitular(clienteCC4);
cc4.deposita(222.0);
```

Não esqueça de importar a classe `Cliente` , `Conta` , `ContaPoupanca` e `ContaCorrente` .

2) Como criaremos outro critério de comparação, adicione outra classe que também implementa a interface `Comparator` . Já implementando os critérios de ordenação, teremos o seguinte:

```
class TitularDaContaComparator implements Comparator<Conta> {

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

3) Agora precisamos criar outro objeto `Comparator` e passar para o método `sort` :

```
lista.sort(new TitularDaContaComparator()); //já deixando mais enxuto
```

4) Para exibirmos as contas e os nomes de seus titulares, faremos o seguinte:

```
for (Conta conta : lista) {
    System.out.println(conta + ", " + conta.getTitular().getNome());
}
```

Alternativamente você pode alterar o método `toString` da classe `Conta` .

5) (Opcional) Teste também a ordenação através da classe `Collections` :

```
Collections.sort(lista)
```