



Laboratório 07 – Interfaces

Atividade **individual**.

1. Crie uma interface e uma classe como as apresentadas a seguir:

Classificavel.java

```
1 public interface Classificavel {  
2  
3     boolean eMenorQue(Classificavel obj);  
4 }
```

Classificador.java

```
1 public class Classificador {  
2  
3     public void ordena(Classificavel[] a) {  
4         Classificavel elem, menor;  
5         int pos;  
6         for (int i = 0; i < a.length - 1; i++) {  
7             elem = a[i];  
8             menor = a[i + 1];  
9             pos = i + 1;  
10            for (int j = i + 2; j < a.length; j++) {  
11                if (a[j].eMenorQue(menor)) { // encontrando o menor  
12                    elemento  
13                        menor = a[j];  
14                        pos = j;  
15                    }  
16                }  
17                if (menor.eMenorQue(elem)) { // troca  
18                    a[i] = a[pos];  
19                    a[pos] = elem;  
20                }  
21            }  
22        }  
23    }
```

Considere agora uma empresa com as classes `Produto`, `Cliente` e `Servico`, que implementam a interface `Classificavel`. Defina atributos e métodos convenientes para essas classes. Inclua ao menos os seguintes atributos: código, para produto; nome, para cliente; e preço, para serviço. Esses serão os atributos passíveis de serem ordenados em cada uma dessas classes. Como exemplo, veja:

Produto.java

```
1 public class Produto implements Classificavel {
2
3     private int codigo;
4
5     public boolean eMenorQue(Classificavel o) {
6         Produto compara = (Produto) o;
7         if (this.codigo < compara.codigo) {
8             return true;
9         } else {
10            return false;
11        }
12    }
13 }
```

Crie uma classe contendo o método `main()`. Defina nesta classe *arrays* (vetores) de clientes, produtos e serviços. Adicione a eles ao menos três elementos; na sequência, ordene esses vetores segundo seus atributos classificáveis e, por fim, imprima seu conteúdo.

Fazer:

- (a) **Modelagem do problema** (diagrama de classes).
- (b) **Código**

Materias complementares (Java)

- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html> (em inglês)
- <https://www.caelum.com.br/apostila/apostila-java-orientacao-objetos.pdf> (em português)