RR2 - 01 Tipos Genéricos e Comparator

Comparator → usamos o método compareTo;
→ se os objetos forem iguais, o método retorna 0;
→ se o objeto for maior do que o passado como parâmetro, temos como retorno um nº > 0;
→ se o objeto for menor do que o passado como parâmetro, temos como retorno um número negativo.

→ exemplo:

```
public class Produto implements Comparable <Produto>

    @Override
    public int compareTo (Produto o){
        return this.codigo - o.getCodigo();
    }
```

↳ usamos o compareTo no equals

→ implementando o método compareTo podemos sobreescrever o método equals.
→ O Comparator é um padrão de projeto que permite que a comparação seja um componente externo.

```
Comparator <Produto> c = new Comparator <Produto> (){

    @Override
    public int compare (Produto o1, Produto o2){
        return o1.getCodigo() - o2.getCodigo();
    }
};
```

→ Podemos ordenar uma lista de objetos usando a comparação:

```
produtos.sort((o1, o2) -> o1.getCodigo() - o2.getCodigo()) // ordem crescente
```

↳ ArrayList produtos = (p2, p1, p5, p4).

```
produtos.sort((o1, o2) -> o2.getCodigo() - o1.getCodigo()) // ordem decrescente
```