

Exercícios sobre Comparable, Comparator e Collections em Java

1. Comparable - Ordem Natural

Crie uma classe Produto com os atributos nome (String) e preco (double).

Implemente a interface Comparable para que os produtos sejam ordenados pelo nome em ordem alfabética.

Crie uma lista com 5 produtos e use Collections.sort(lista) para ordená-los.

Imprima a lista antes e depois da ordenação.

2. Comparable - Ordenando por número

Modifique a classe Produto do exercício anterior para que a ordenação natural seja pelo preco, do menor para o maior.

Use novamente Collections.sort(lista) e mostre a lista ordenada.

3. Comparator - Ordem personalizada

Crie um Comparator para ordenar os produtos do exercício anterior em ordem decrescente de preço.

Use Collections.sort(lista, novoComparator) para aplicar a ordenação.

4. Comparator com Classe Anônima

Sem criar uma nova classe, use uma classe anônima para ordenar a lista de produtos pela quantidade de letras no nome.

5. Comparator.comparing (Java 8+)

Use Comparator.comparing() para ordenar uma lista de objetos Pessoa pela idade.

Dica: Pessoa deve ter os atributos nome e idade.

6. Collections.max() com Comparator

Crie uma lista de contas bancárias com atributo saldo.

Use Collections.max(lista, Comparator.comparing(Conta::getSaldo)) para encontrar a conta com o maior saldo.

Imprima os dados da conta com maior saldo.

Exercícios sobre Comparable, Comparator e Collections em Java

7. Desafio final

Crie uma classe Aluno com atributos nome, nota1, nota2.

Implemente Comparable para ordenar os alunos pela média das notas (ordem decrescente).

Crie uma lista com 5 alunos, ordene-a e imprima os dados.