

# Linguagem de Programação 1

## Coleções

Prof. Fábio José Rodrigues Pinheiro

NOVEMBRO DE 2021



**INSTITUTO  
FEDERAL**

Catarinense

---

Campus  
Videira

# APIs Java: Coleções

- Em Java, coleção é um objeto que agrupa múltiplos elementos dentro de uma única unidade
  - Usadas para armazenar, obter e manipular dados agregados
- Representam itens que formam um grupo natural
  - Baralho, pasta de e-mails, catálogo telefônico
- O *Java Collections Framework* provê também algoritmos para busca e ordenação em coleções



## ■ Set

- Coleção que não permite elementos duplicados

## ■ List

- Coleção ordenada de elementos e permite elementos duplicados

## ■ Queue

- Fila que ordena elementos para serem processados posteriormente, por exemplo, FIFO

## ■ Map

- Mapeia chaves para valores. Não permite chaves duplicadas e cada chave pode mapear somente um valor



# Listas – ArrayList

```
1 ArrayList<String> produtos = new ArrayList<>();
2 int n = produtos.size(); // obtendo o total de elementos na coleção
3
4 produtos.add("Livro");
5 produtos.add("Jogo");
6
7 String item = produtos.get(1); // obtém elemento na posição 1
8
9 String s = produtos.toString(); // [Livro, Jogo]
10
11 boolean b = produtos.contains("Jogo"); // verificando se existe
12
13 Collections.sort(produtos); // ordena todos elementos da lista
14
15 produtos.remove(1); // removendo elemento
16 //ou
17 produtos.remove("Livro");
18
19 produtos.clear(); // remove todos elementos da lista
```



# Percorrendo uma ArrayList

```
1 // Percorrendo e alterando um elemento específico
2 for(String elemento: produtos){
3     if (elemento.equals("Jogo")){
4         int posicao = produtos.indexOf(elemento);
5         produtos.set(posicao, "Game");
6     }
7 }
8 //Iterando uma coleção
9 for(String elemento: produtos){
10     System.out.println(elemento);
11 }
12 //Iterando com lambda
13 produtos.forEach(elemento->System.out.println(elemento));
```



# HashMap

```
1 Map<String, String> cores = new HashMap<>();
2 cores.put("Vermelho", "FF0000");
3 cores.put("Verde", "00FF00");
4 cores.put("Azul", "0000FF");
5
6 // Percorrendo as chaves
7 for (String chave : cores.keySet()) {
8     System.out.println(chave);
9 }
10
11 // Percorrendo os valores
12 for (String valor : m.values()) {
13     System.out.println(valor);
14 }
15 // Percorrendo um HashMap
16 for (Map.Entry<String, String> elemento: cores.entrySet()){
17     System.out.println(elemento.getKey() + ":" + elemento.getValue());
18 }
```



- Escreva a classe `ListaDePalavras` que represente uma lista de palavras (strings), de forma que seja possível representar palavras repetidas na lista.
  - Escreva um método **conta** que recebe como argumento uma string e retorna o número de ocorrências dessa string na lista (retornar zero caso não exista)





- Faça uma classe que represente os estados do Brasil através do par <Sigla, Nome>, utilizando uma das Coleções apresentadas:
  - Crie um método para inserir os dados na Coleção
  - Crie um método que retorne o nome do estado a partir da Sigla
  - Crie um método para imprimir todos os estados e suas Siglas
  - Defina os membros desta classe como estáticos, se achar que é possível.



- Implemente uma classe Java para um Carrinho de Compras.
  - Essa classe deve possuir uma coleção de itens:
    - Item deverá ser uma classe que representa um item de compra que deve possuir um produto e uma quantidade
    - Produto deverá também ser uma classe com os seguintes atributos: código, nome, preço.
  - A classe Carrinho deve ter métodos para incluir e remover um item da coleção.
  - Deve ter ainda um método para listar os itens do carrinho e o total a pagar, calculado pela soma dos preços de todos os produtos do carrinho.



- <https://docs.oracle.com/javase/tutorial/essential/exceptions>
- <https://docs.oracle.com/javase/tutorial/collections/>
- <https://docs.oracle.com/javase/tutorial/java/data/buffers.html>
- <https://docs.oracle.com/javase/8/docs/technotes/guides/language/foreach.html>
- <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>
- <https://blog.caelum.com.br/java-8-lambda-ou-method-reference-entenda-a-diferenca/>
- <https://docs.oracle.com/javase/8/docs/api/javax/swing/text/MaskFormatter.html>

