

# Programar em Linguagem Orientada a Objetos Básica

Prof. Arthur M. Araújo

# Introdução às Java Collections

- Conceito
- Arrays
- Collections
  - ArrayList
  - HashSet
  - HashMap
- Operações Comuns em Coleções
- Exercício



## Conceito de Coleções

**Collections** em Java são estruturas de dados que permitem armazenar e organizar grupos de dados de maneira eficiente. Elas fornecem interfaces e implementações para trabalhar com **listas** de elementos, **conjuntos** sem duplicatas e **mapas** associativos (pares chave-valor). As collections facilitam a **manipulação, pesquisa, inserção e remoção** de elementos em conjuntos de dados dinâmicos, oferecendo uma variedade de métodos úteis para operações comuns.

Em resumo, collections são ferramentas poderosas que simplificam a manipulação de conjuntos de dados em Java, tornando o processo mais flexível e eficiente.

# Arrays

**Arrays** em Java são estruturas de dados que armazenam elementos do mesmo tipo em uma **sequência fixa**. Cada elemento é identificado por um **índice**, começando do **zero**. Os arrays possuem um tamanho **fixo** durante toda a execução do programa, e você pode **acessar**, **modificar** e **percorrer** seus elementos usando índices.

Adicionando o nome **João** na posição “0” e **José** na posição “1” do array de nomes.

Exemplos:

```
String[] arrayDeNomes = new String[2];  
arrayDeNomes[0] = “João”  
arrayDeNomes[1] = “José”
```

Estrutura do  
Array em Java

## Collections - ArrayList

ArrayList em Java é uma estrutura de dados que fornece uma coleção dinâmica, eficiente e fácil de usar para armazenar e manipular elementos, sem as limitações de tamanho fixo encontradas nos arrays tradicionais.

```
8      // Declarando e inicializando um ArrayList
9      List<String> frutas = new ArrayList<>();
10     frutas.add("Maçã");
11     frutas.add("Banana");
12     frutas.add("Morango");
13
14     // Acessando elementos por índice
15     String primeiraFruta = frutas.get(0);
16
17     // Removendo um elemento
18     frutas.remove(0: "Banana");
```

## Collections - HashSet

**HashSet** em Java é uma implementação da interface Set que representa uma coleção de elementos **únicos**, onde não são permitidos elementos duplicados.

```
10 // Declarando e inicializando um HashSet
11 Set<Integer> numeros = new HashSet<>();
12 numeros.add(10);
13 numeros.add(5);
14 numeros.add(10); // Ignorado, pois 10 já existe no conjunto
15
16 // Verificando se um elemento existe
17 boolean contemCinco = numeros.contains(5);
18
19 // Removendo um elemento
20 numeros.remove(10);
```

## Collections - HashMap

**HashMap** em Java é uma estrutura de dados que permite associar **chaves** a **valores**, proporcionando uma rápida recuperação de valores com base nas **chaves**, independentemente do tamanho da coleção.

```
7      // Declarando e inicializando um HashMap
8      Map<String, Integer> idades = new HashMap<>();
9      idades.put("João", 25);
10     idades.put("Maria", 30);
11
12     // Acessando um valor pela chave
13     int idadeJoao = idades.get("João");
14
15     // Removendo um par chave-valor
16     idades.remove(key: "Maria");
```

# Operações Comuns em Coleções

- `size();`
- `isEmpty();`
- `contains(objeto);`
- `remove(objeto);`

```
7 // Exemplo com operações comuns
8 List<String> lista = new ArrayList<>();
9 lista.add("Item 1");
10 lista.add("Item 2");
11
12 // Verificando o tamanho da lista
13 int tamanhoLista = lista.size();
14
15 // Verificando se a lista está vazia
16 boolean listaVazia = lista.isEmpty();
17
18 // Verificando se um item está na lista
19 boolean contemItem = lista.contains("Item 2");
20
21 // Removendo um item
22 lista.remove("Item 1");
```



## Exercício - Gerenciador de Tarefas

Agora vamos aplicar o que aprendemos sobre **coleções**. Criaremos um programa para **gerenciar as tarefas** uma pessoa.

Link da atividade: [Desafio - Gerenciador de Tarefas](#)

**Observação:** A atividade solicita o uso da coleção **ArrayList**, mas recomendo refazê-la utilizando outras coleções que estudamos, como **HashSet** e **HashMap** para que entendam na prática como que cada uma funciona.

# Obrigado!

## Contatos



**Prof. Arthur M. Araújo**



**[arthur.araujo@maisunifacisa.com.br](mailto:arthur.araujo@maisunifacisa.com.br)**