

**INSTITUTO FEDERAL**  
Goiás

Instituto Federal de Goiás  
Câmpus Goiânia

Bacharelado em Sistemas de Informação  
Disciplina: Programação Orientada a Objetos I

# Trabalhando com Coleções de Dados

Prof. Ms. Dory Gonzaga Rodrigues  
Goiânia - GO

# Collection

Uma coleção é simplesmente um objeto que agrupa vários elementos em uma única unidade ou recipiente (variável)

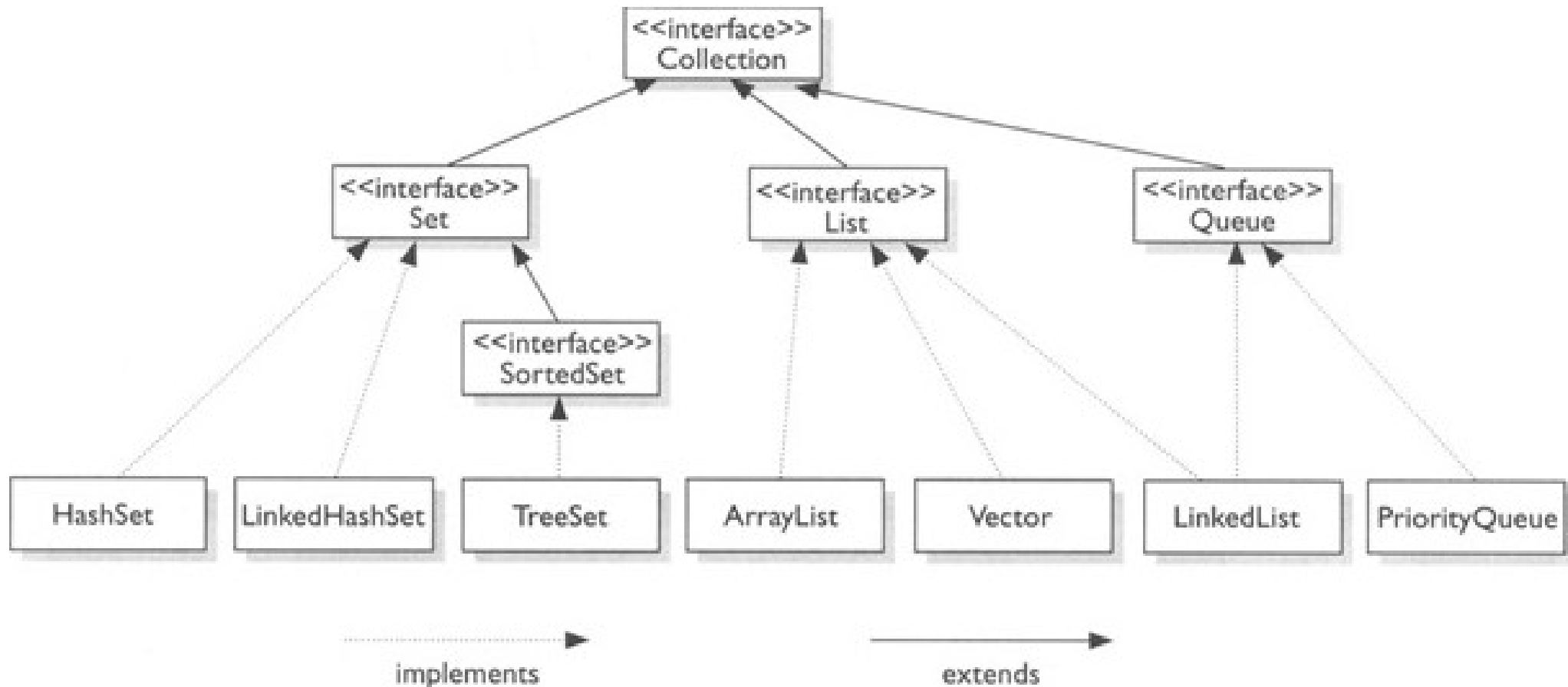


**As coleções são usadas para:**

- ▶ armazenar, recuperar, manipular dados agregados.
- ▶ Representam itens de dados que formam um grupo natural:
  - \* uma coleção de cartas
  - \* uma lista de alunos
  - \* uma lista de produtos de uma venda

# Collection

A linguagem java disponibiliza **INTERFACES** e **CLASSES** dessa estrutura de coleções através do pacote **java.util**.



# Interface Collection

**Esta é a interface raiz do framework**

Serve de referência para as implementações das coleções.



**Toda a classe que implementa a interface Collection tem os seguintes métodos**

▶ **add( E e )**

Adiciona o objeto passado por parâmetro a coleção.

▶ **addAll( Collection<? extends E> c )**

Adiciona todos os elementos de uma coleção (passada por parâmetro) à coleção que chamou o método.

▶ **remove(Object o )**

Remove o objeto passado por parâmetro da coleção.

**CONTINUA >>**

# Interface Collection



**Toda a classe que implementa a interface Collection tem os seguintes métodos**

▶ **removeAll(Collection<?> c)**

Remove todos os elementos de uma coleção (passada por parâmetro) da coleção que chamou o método.

▶ **contains(Object o)**

Retorna true se o objeto passado por parâmetro estiver dentro da coleção, senão retorna false.

▶ **isEmpty()**

Retorna true se a coleção estiver vazia.

▶ **Size()**

Retorna o número de objetos da coleção.

# Interface Collection



**Toda a classe que implementa a interface Collection tem os seguintes métodos**

- ▶ **Clear()**

Remove todos os objetos de dentro da coleção.

- ▶ **ToArray()**

Retorna um array contendo todos os objetos da coleção.

- ▶ **toArray( T[] a)**

Retorna um array contendo todos os objetos da coleção. Se os objetos da coleção couberem no array passado por parâmetro este será retornado, senão um novo array será criado e retornado.

- ▶ **Iterator()**

Retorna um objeto Iterator utilizado para percorrer os objetos da coleção. Este método é especificado na interface Iterable a qual Collection estende.

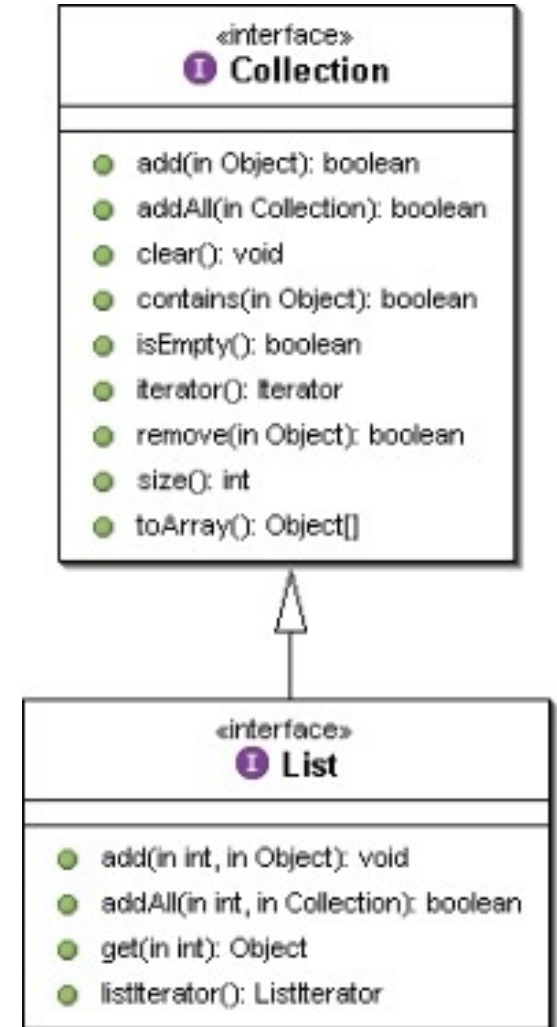
# ArrayList

Esta é a classe que implementa uma coleção utilizando a estrutura de VETOR



## ArrayList:

- ▶ Implementa os métodos da interface List
- ▶ Como a interface List herda da interface Collection todos os métodos de Collection também estão implementados no ArrayList



# Como criar uma variável do tipo ArrayList ?

Abaixo temos a linha de código padrão para criar e instanciar uma variável que seja uma coleção de objetos

```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class TesteAluno {
5
6     public static void main(String[] args) {
7
8         Aluno a1 = new Aluno("Ana");
9         Aluno a2 = new Aluno("Pedro");
10        Aluno a3 = new Aluno("Maria");
11
12        List <Aluno> listaDeAlunos = new ArrayList<>();
13    }
```

Interface Collection

Tipo de objeto que será armazenado

Classe que implementa a interface Collection