

**MODULO 5**  
**Coleções****Classe Empregado**

```
class Empregado {
    private String nome;
    private double salario;

    public Empregado(String nome, double salario) {
        this.nome = nome;
        this.salario = salario;
    }

    public Empregado(String nome) {
        this.nome = nome;
        this.salario = 5000;
    }

    public void trabalhar(){
        System.out.println("Eu trabalho muito");
    }

    public double calculaBonificacao(){
        return 0.2*salario;
    }

    public double calculaBonificacao(double percentagem) {
        return percentagem*salario;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }
}
```

**Classe Sindicato**

```
public class Sindicato {
    private ArrayList<Empregado> lstEmpregados;
    private String nomeSindicato;
    private Empregado presidente;

    public Sindicato(){
        lstEmpregados = new ArrayList<>();
    }

    public Sindicato(String nomeSindicato, Empregado presidente){
        lstEmpregados = new ArrayList<>();
        this.presidente=presidente;
        lstEmpregados.add(presidente);
    }

    public void incluirEmpregado(Empregado emp) {
        lstEmpregados.add(emp);
    }

    public void excluirEmpregado(Empregado emp) {
        lstEmpregados.remove(emp);
    }

    public void excluirTodosOsEmpregados(Empregado emp) {
        lstEmpregados.removeAll(lstEmpregados);
    }

    public int getTotalEmpregados() {
        return lstEmpregados.size();
    }

    public void imprimirEmpregados() {
        for (Empregado empregado : lstEmpregados) {
            System.out.println("- " + empregado.getNome());
        }
    }
}
```

Para testar a Classe Empregado, criamos outra classe

### Classe SindicatoTeste

```
public class SindicatoTeste {  
  
    public static void main(String [] args){  
        Empregado presidente = new Empregado("Joao Ferreira",35);  
        Sindicato sindicato = new Sindicato("Os empregados  
felices", presidente);  
        sindicato.incluirEmpregado(new Empregado("Marcio"));  
        sindicato.incluirEmpregado(new Empregado("Leandro"));  
        sindicato.incluirEmpregado(new Empregado("Fulgencio"));  
        System.out.println("O número de empregados é: " +  
sindicato.getTotalEmpregados());  
        sindicato.imprimirEmpregados();  
    }  
}
```

Saída:

```
O número de empregados é: 4  
- Joao Ferreira  
- Marcio  
- Leandro  
- Fulgencio
```

**Lista de Exercícios 5**

1. Escreva uma classe chamada **BlocoNotas** que siga a especificação abaixo:

Atributos

- notas (conjunto de notas de tipo ArrayList).

Métodos

- Implemente um método construtor que inicialize o atributo notas.
- Implemente os métodos a seguir:

Método	Parâmetros	Descrição
salvarNota	String nota	Permite armazenar notas no objeto de tipo ArrayList.
imprimirNota	int i	Mostra na tela uma nota baseado no índice i da nota (verifica se o índice é válido, caso contrário imprime mensagem de invalidade).
imprimirNotas		Mostra todas as notas armazenadas atualmente.
apagarNota	int i	Permite apagar uma nota dada no índice i (verifica se o índice é válido, caso contrário imprime mensagem de invalidade).
getTotalNotas		Mostra o número de notas armazenado

\*\* Na implementação dos métodos, deve-se verificar que o número recebido para ser alocado na variável num seja positivo.

Testando:

- Escreva uma classe de teste chamada BlocoNotasTeste que demonstra as capacidades da classe BlocoNotas.
- Na classe BlocoNotasTeste crie um método main. Dentro do main, crie um bucle que insira 20 notas diferentes com qualquer mensagem. Após mostre todas as notas (método imprimirNotas).
- Finalmente apague uma nota qualquer e mostre as notas (método imprimirNotas) e o número de notas (getTotalNotas).

2. Escreva as classes especificadas a seguir:

- a) Escreva uma classe chamada **Livro** que siga a especificação abaixo

Atributos

- titulo (tipo String).
- autor (tipo String).
- numPaginas (tipo int).

Métodos

- Implemente um método construtor.
- Implemente os métodos get e set
- Implemente o método mostrarLivro que exiba o título, autor e numero de páginas do livro.

b) Implemente uma classe **Biblioteca** que siga a especificação abaixo:

Atributos

- livros (conjunto de objetos de tipo Livro armazenados em um ArrayList).

Métodos

- Implemente um método construtor que inicialize o atributo livros.
- Implemente os métodos a seguir:

Método	Parâmetros	Descrição
adicionarLibro	Livro livro	Permite armazenar o livro no objeto de tipo ArrayList.
mostrarLivro	int i	Mostra na tela os atributos de um livro (chama ao método mostrarLivro da classe Livro) baseado no índice i (verifica se o índice é valido, caso contrario imprime mensagem de invalidade).
mostrarLivro	Livro li	Mostra na tela os atributos de um livro li especifico (chama ao método mostrarLivro da classe Livro).
mostrarLivros		Mostra todos os livros armazenados atualmente (chama ao método mostrarLivro da classe Livro).
apagarLivro	int i	Permite apagar um livro dado no indice i (verifica se o indice é valido, caso contrario imprime mensagem de invalidade).
apagarLivro	Livro li	Permite apagar um livro especifico dado referenciado pelo objeto li.
getTotalLivros		Mostra o número de livros armazenados.

Testando:

- Escreva uma classe de teste chamada BibliotecaTeste que demonstra as capacidades da classe Biblioteca.
- Na classe BibliotecaTeste crie um método main. Dentro do main, crie um objeto de tipo Biblioteca e use todos os métodos disponíveis na classe e imprima as saidas.