



# Programação Orientada a Objetos I

**Prof<sup>a</sup>. Angela Abreu Rosa de Sá, Dr<sup>a</sup>.**

---

*Contato: [angelaabreu@gmail.com](mailto:angelaabreu@gmail.com)*

# Material Didático

## Programação Orientada a Objetos

### Sumário

Unidade 1   Fundamentos da orientação a objetos .....	7
Seção 1.1 - Histórico e introdução à orientação a objetos .....	9
Seção 1.2 - Conceitos básicos de orientação a objetos .....	22
Seção 1.3 - Construtores e sobrecarga .....	37
Unidade 2   Estruturas de programação orientadas a objetos .....	59
Seção 2.1 - Estruturas de decisão e controle em Java .....	61
Seção 2.2 - Estruturas de repetição em Java .....	76
Seção 2.3 - Reutilização de classes em Java .....	93
Unidade 3   Exceções, classes abstratas e interfaces .....	111
Seção 3.1 - Definição e tratamento de exceções .....	113
Seção 3.2 - Definição e uso de classes abstratas .....	126
Seção 3.3 - Definição e uso de interfaces .....	141
Unidade 4   Aplicações orientadas a objetos .....	155
Seção 4.1 - Arrays em Java .....	157
Seção 4.2 - Strings em Java .....	173
Seção 4.3 - Coleções e arquivos .....	188

# Orientação a objetos

**Encapsulamento**

**Herança**

**Composição**

**Polimorfismo**

**Princípio da abstração**

# Conceitos Fundamentais

---

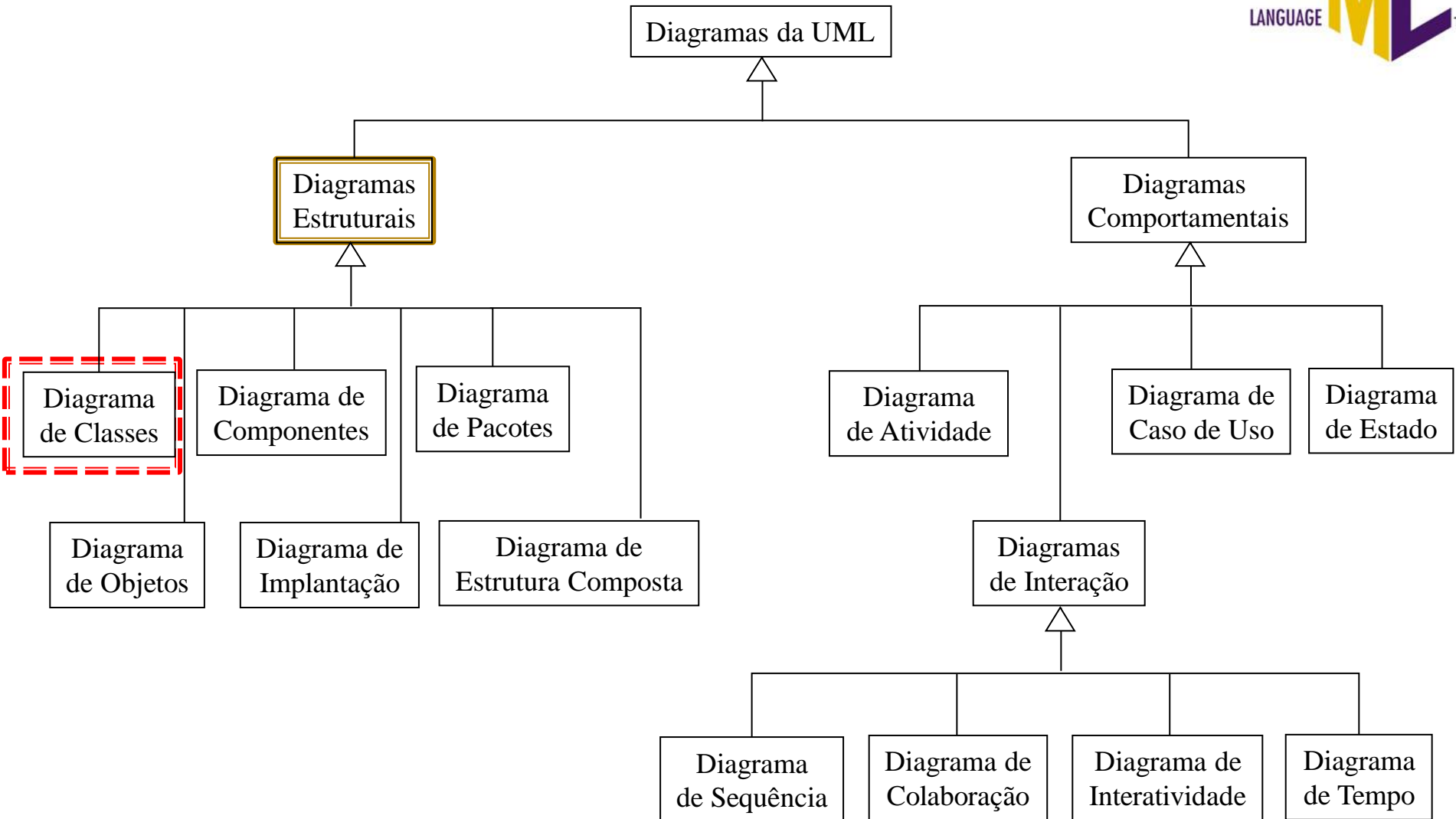
- Classe / Objeto
- Construtor
- Atributos
- Métodos
- Sobrecarga
- Encapsulamento
- Herança/Generalização/Especialização
- Polimorfismo

# Modelagem



- Linguagem de Modelagem que **permite a representação de conceitos do mundo real**, sob a ótica da orientação a objetos.
- A UML permite que desenvolvedores **visualizem** os produtos de seus trabalhos em **diagramas padronizados**

# Diagramas da UML

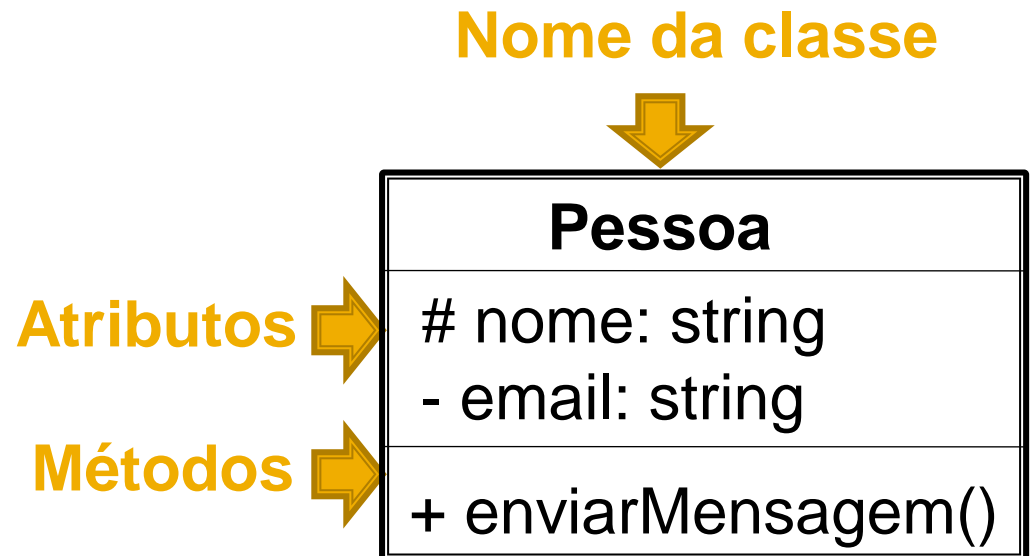


# Classe

- Uma classe é representada por um retângulo com **três divisões**:

- Nome da Classe
- Atributos da Classe
- Métodos da Classe

Nome da classe
Lista de atributos
Lista de métodos



# Visibilidade

---

- **Pública (+)**

- O atributo ou método pode ser utilizado por qualquer classe

- **Protegida (#)**

- Somente a classe ou sub-classes terão acesso

- **Privada (-)**

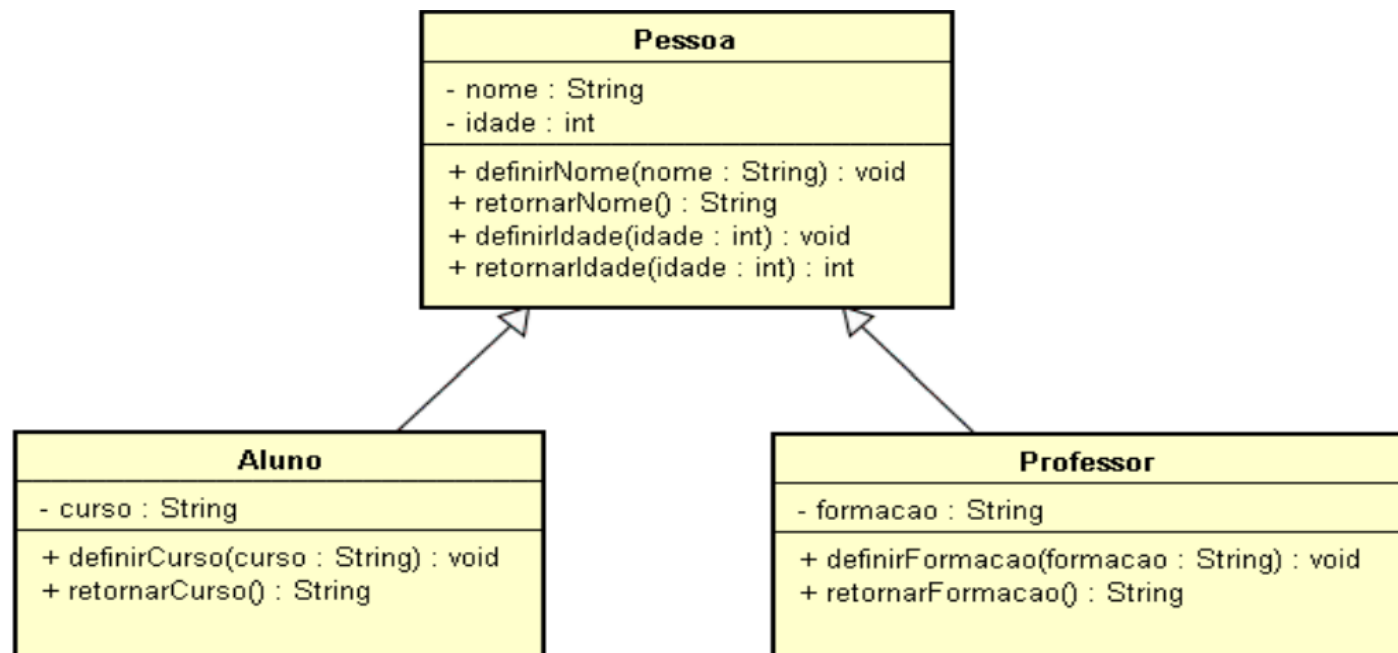
- Somente a classe terá acesso

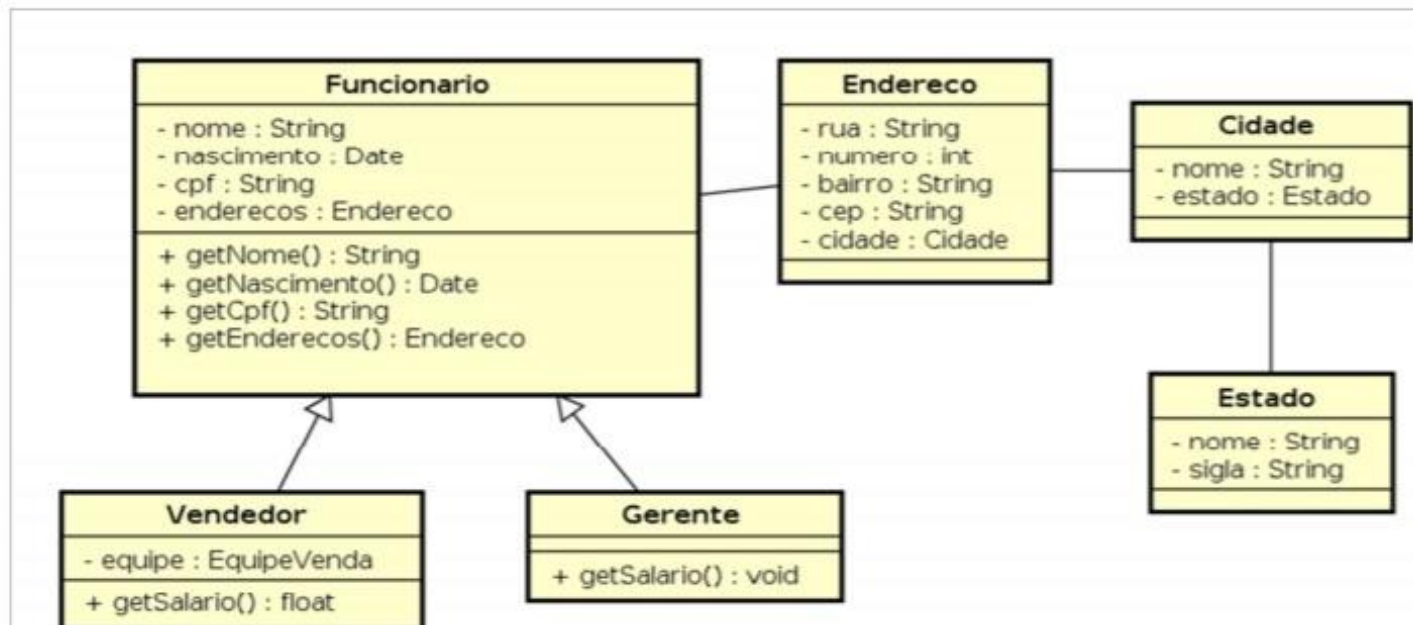


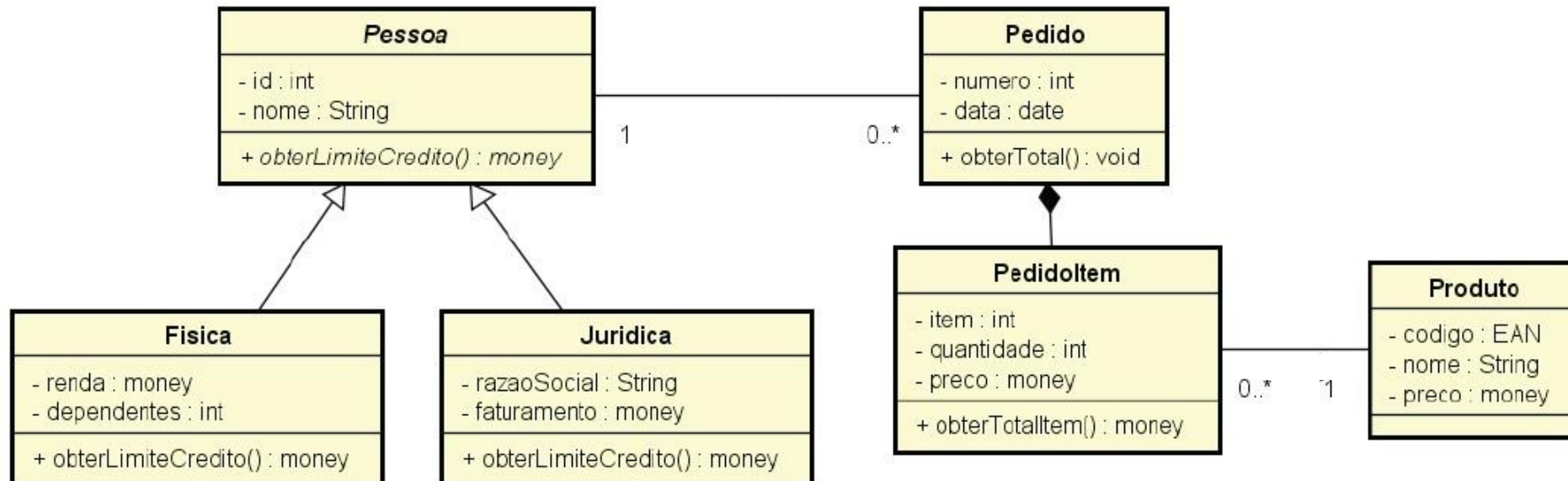
# Relacionamento

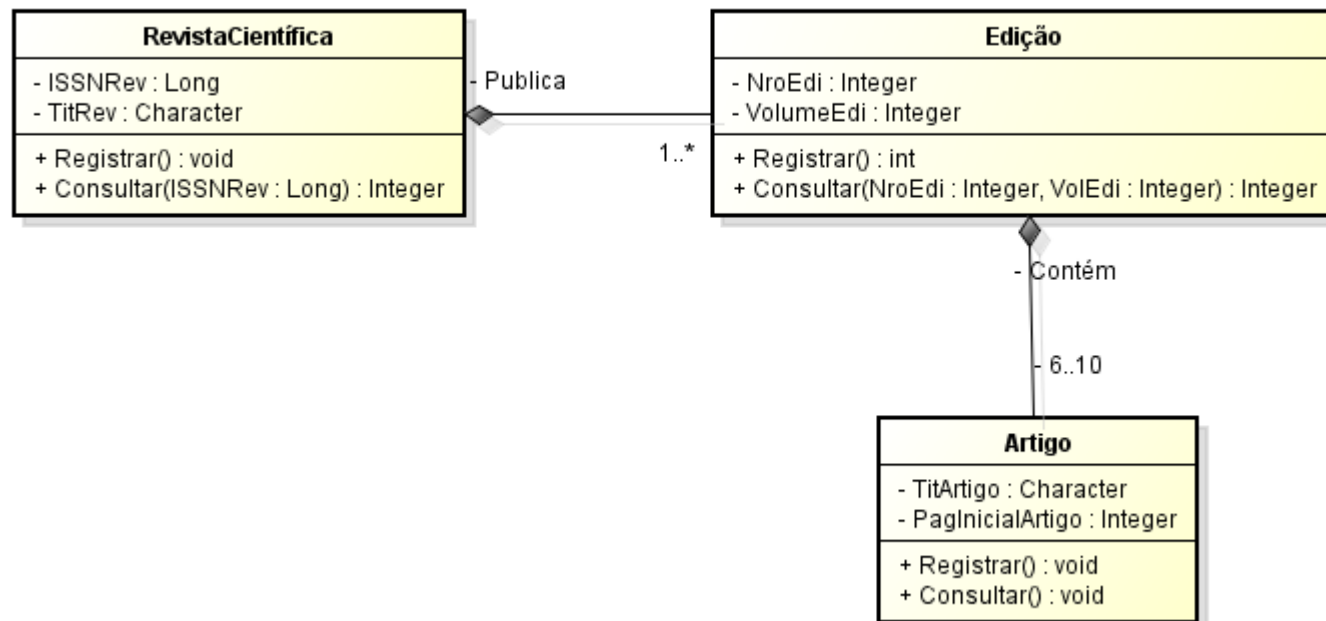
---

- Classes possuem **relacionamentos** entre elas
  - Compartilham informações
  - Colaboram umas com as outras
- Principais tipos de relacionamentos
  - **Associação**
  - **Agregação / Composição**
  - **Herança**







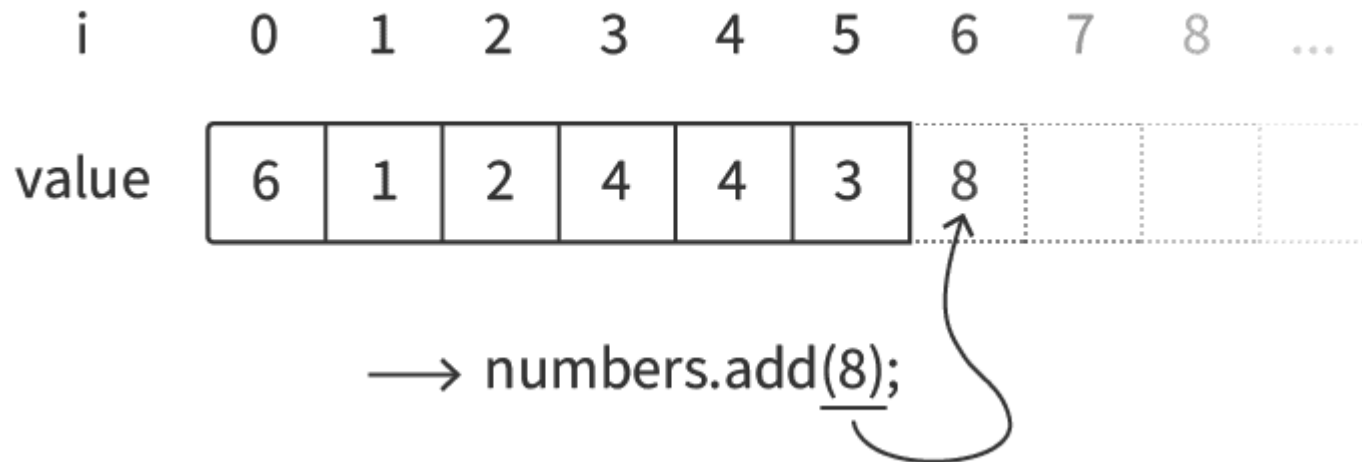


# ArrayList

```
import java.util.ArrayList;
```

São Arrays| Vetores que crescem **dinamicamente**: não precisamos especificar o tamanho!

```
ArrayList<INTEGER> numbers = new ArrayList<INTEGER>();
```



```
public class Edicao {  
  
    private int Numero;  
  
    private ArrayList<Artigo> Array_Artigo = new ArrayList<Artigo>(); //Cria um arraylist de objetos Artigo  
  
    public Edicao(int Numero)  
    {  
        this.Numero = Numero;  
    }  
  
    public void InsereArtigo(String Titulo)  
    {  
        Artigo objArtigo = new Artigo( Titulo);  
        Array_Artigo.add(objArtigo);  
    }  
  
    public ArrayList<Artigo> getArray_Artigo() {  
        return Array_Artigo;  
    }  
  
    public int getNumero() {  
        return Numero;  
    }  
}
```

```
System.out.print("\n ### CONTEUDO DA REVISTA #####");  
//mostrar todas as edições da revista e todos os artigo de cada edição  
for(Edicao edicao: objetoRevista.getArray_Edicao())  
{  
    // mostrar o número da edição  
    System.out.print("\n ----- Edição: " + edicao.getNumero() );  
  
    //mostrar todos os artigo da edição  
    for(Artigo artigo: edicao.getArray_Artigo())  
    {  
        // mostrar o título do artigo  
        System.out.print("\n ** Artigo: " + artigo.getTitulo());  
    }  
}
```

```
}
```

```
}
```



## FEVEREIRO / 2023

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20 <sup>N</sup>	21 <sup>N</sup>	22 <sup>M</sup>	23 <sup>C</sup>	24	25
26	27	28				

### FEVEREIRO

- 01. Início do Período Letivo
- 13 a 15. Semana Acadêmica
- 20 a 21. Carnaval
- 23. Início das Aulas

## MARÇO / 2023

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

### MARÇO

## ABRIL / 2023

D	S	T	Q	Q	S	S
						1
2	3	4	5	6 <sup>M</sup>	7 <sup>N</sup>	8 <sup>M</sup>
9	10	11	12	13	14	15 <sup>M</sup>
16	17	18	19	20	21 <sup>N</sup>	22 <sup>M</sup>
23	24	25	26	27	28	29
30						

### ABRIL

- 07. Paixão de Cristo
- 10 a 14. Avaliação Oficial - 1º Bimestre
- 21. Tiradentes
- 28. Lançamento das notas - 1º Bimestre

Avaliação 1

## MAIO / 2023

D	S	T	Q	Q	S	S
	1 <sup>N</sup>	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

### MAIO

- 01. Dia do Trabalho

## JUNHO / 2023

D	S	T	Q	Q	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24 <sup>I</sup>
25	26	27	28	29	30	

### JUNHO

- 01 a 07. Avaliação Oficial - 2º Bimestre
- 05 a 17. Avaliação Oficial DI
- 08. Corpus Christi

## JULHO / 2023

D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

### JULHO

- \*Poderá haver atividades ou compensação de horas, exceto em caso de férias ou recessos previstos em acordos sindicais.

# Cronograma

---

- **28 Março:** Lista de exercícios
- **04 Abril:** Dúvidas
- **11 Abril:** Avaliação 1

**Cla:** Em todos os programas, não se esqueça de criar uma interface adequada para o usuário.

1) Escreva em Java a classe Contador, que encapsule um valor usado para contagem de eventos (incrementar). A classe deve oferecer métodos que devem:

- Zerar;
- Incrementar;
- Retornar o valor do contador.

Implemente o programa principal que irá instanciar um objeto da classe Contador. O programa deverá solicitar ao usuário qual é a operação que será realizada: 1) Incrementar contador; 2) Zerar Contador; 3) Mostrar o valor do contador.

2) Construa a classe Livro em Java, que obedeça à descrição abaixo:



- Crie os métodos `get` e `set` para cada um dos atributos.
- Crie ainda o método `verificarProgresso` que deverá calcular a porcentagem de leitura do livro até o momento. Para isso, deverá usar os valores dos atributos `qtdPaginas` e `paginaLida`, através da fórmula:  $porcentagem = \frac{paginaLida}{qtdPaginas} * 100$ . O valor da porcentagem deverá ser mostrado na tela conforme a mensagem "Você já leu X por cento do livro", onde o valor de X é o valor calculado pela fórmula apresentada anteriormente.

Crie um programa principal que instancie um objeto da Classe Livro. Em seguida, mostre os dados do livro cadastrado e o progresso de leitura. Altere o título e a quantidade de páginas lidas. Apresente novamente os dados do livro e o progresso de leitura.

3) Utilizando o paradigma Orientado a Objetos na linguagem Java, implemente a classe Universidade que tem as seguintes características:

- Atributos:** - nome da universidade
- Métodos:** - Retornar o nome da Universidade;
- Construtor;

Em seguida, implemente a classe Pessoa. Esta classe deverá contemplar:

- Atributos:** dia, mês e ano de nascimento, idade, nome da pessoa e a universidade em que ela trabalha (objeto da Classe Universidade).
- Métodos:**
  - Construtor;
  - Calcular a idade da pessoa (*clã: considere apenas o ano de nascimento para o cálculo da idade*);
  - Retornar o valor da idade;
  - **Renovar** o nome da pessoa;
  - Retornar o nome da Universidade em que a pessoa trabalha.

Por fim, o programa principal deverá:

- Criar dois objetos da classe Pessoa: um representando Albert Einstein (nascido em 14/3/1879) e o outro representando Isaac Newton (nascido em 4/1/1643). Einstein trabalhou como professor de física em Princeton (Nova Jersey - Estados Unidos da América). Newton trabalhou como professor de matemática em Cambridge (Inglaterra).
- Apresentar um relatório completo das pessoas cadastradas, contendo todos os dados, inclusive idade e o nome da Universidade em que cada um trabalha.

4) Utilizando a linguagem Java, implemente um programa que irá cadastrar um Assess de professor e outro Assess de coordenador, conforme a estrutura descrita no Diagrama de Classes abaixo.



O programa principal deverá ter um menu para **solicitar que o usuário digite as informações | abrir um relatório com todas as dados cadastrados de todos os Professores e Coordenadores | Sair**

5) Construa a classe Contato em Java, que obedeça à descrição abaixo:

- A classe deve possuir dois atributos privados: nome (do tipo String) e telefone (do tipo String)
- Crie um construtor para a classe que recebe 2 parâmetros, cada um representando um dos atributos da classe.
- Crie os métodos de acesso (get e set) para os atributos da classe

Contato
- nome: String - telefone: String
+ getNome(): String + setNome(String): void + getTelefone(): String + setTelefone(String): void + Contato(String, String)

Em seguida, construa a classe Agenda, a qual deve obedecer a descrição abaixo:

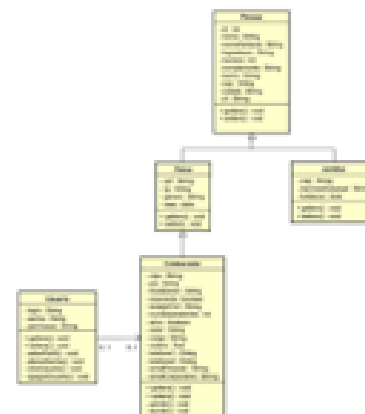
- Crie um método main na classe
- Crie um vetor de objetos Contato do tipo ArrayList (ex: ArrayList<Contato> agendaDeContatos)
- Elabore um menu na tela que ofereça as opções abaixo. O seu programa deve ficar executando enquanto o usuário não digitar a opção 4(sair). Solicite que o usuário informe a opção desejada:

- Na opção **Cadastrar Contato** (opção 1), solicite que o usuário digite o nome e o telefone do contato que ele deseja inserir. Crie um objeto do tipo Contato com as informações digitadas pelo usuário e insira o objeto no vetor. .
- Na opção **Buscar Contato** (opção 2), solicite que o usuário informe o nome do contato que ele deseja buscar. Percorra o vetor de objetos de forma a verificar se o contato que o usuário deseja buscar existe na coleção. Caso exista, escreva na tela o número de telefone do contato. Se não existir, exiba a mensagem "contato inexistente".
- Na opção **Imprimir Agenda** (opção 3), escreva na tela as informações nome e telefone de todos os contatos que existem no vetor.
- Encerre a execução somente quando a opção **Sair** (opção 4) for digitada.

6) Implemente um programa para fazer um cadastro completo dos vendedores e do gerente de um Empresa de Equipamentos Tecnológicos - **utilizar ARRAYLIST**. Ao final do cadastro, o programa deverá apresentar um relatório completo de todas as informações dos funcionários.



7) Implemente um programa para cadastrar as pessoas (usuário ou pessoa jurídica) que trabalham na Faculdade de Computação **utilizar ARRAYLIST**. Ao final do cadastro, o programa deverá apresentar um relatório completo de todas as informações dos funcionários.





# Muito Obrigada!

**Prof<sup>a</sup>. Angela Abreu Rosa de Sá, Dr<sup>a</sup>.**

---

*Contato: [angelaabreu@gmail.com](mailto:angelaabreu@gmail.com)*