**Curso em vídeo – POO Em Java**

- A **classe** é como se fosse o molde do objeto;

- **Atributos:** Características do objeto;

- **Métodos**: Ações/comportamento do objeto;

- **Estado:** Características atuais, isto é, no momento em que eu estou analisando esse objeto;

- **Dica:**

Quando eu for criar uma classe, eu devo pensar em tudo que o objeto tem, por exemplo: A classe caneta tem um \*modelo\* do tipo caractere, \*cor\* do tipo caractere, \*ponta\* do tipo real, \*carga\* do tipo inteiro e ela pode ser \*tampada\*, logo é do tipo lógico...



- Instanciar uma classe é a mesma coisa que criar um objeto

**- Por exemplo:**

Caneta\_1 = nova Caneta (“Caneta\_1 vira um objeto após o “nova”, pois com esse termo, objetos são criados/instanciados)  
  
obs.: O que vem depois do nova é a classe que eu vou utilizar, o que vem antes é o nome do objeto que vai existir.  
  
- Se eu quiser que essa mesma caneta seja azul, tenha a ponta 0.5, esteja destampada e rabisque é só eu fazer o seguinte:  
  
Caneta\_1.cor = ”Azul”  
Caneta\_1.ponta = 0.5

Caneta\_1.tampada = falso

Caneta\_1.rabiscar()   
  
Obs.: Perceba que aqui o rabiscar tem parênteses no final, isto porque “rabiscar” é um método, já se não tiver parênteses é um atributo.  
  
- Se eu quiser criar outra caneta dessa mesma classe (Mesmo molde) eu posso! Por exemplo, vou criar uma caneta de cor vermelha, ponta 1.0 e destampada também, veja só:  
  
Caneta\_2 = nova Caneta Criei a segunda caneta;

Caneta\_2.cor = “Vermelho” Defini a cor da tinta como vermelho;  
Caneta\_2.ponta = 1.0 Defini que a ponta dessa caneta será 1.0;  
Caneta\_2.tampada = falso Defini que ela está destampada;  
Caneta\_2.tampar() Destampei ela novamente;  
  
P.S: Lembrando que as duas canetas são da mesma classe, isto é, elas tem as mesmas características (Atributos), porém tem valores ou estados diferentes.

**- Definição:**

- Classe: Define os atributos e os métodos comuns que serão compartilhados por um objeto.  
  
- Objeto: É a instância (criação) de uma classe.

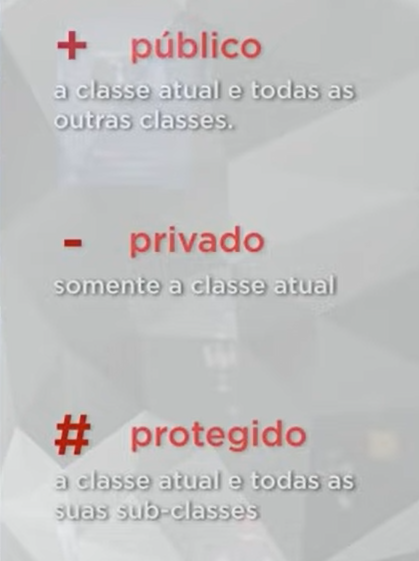
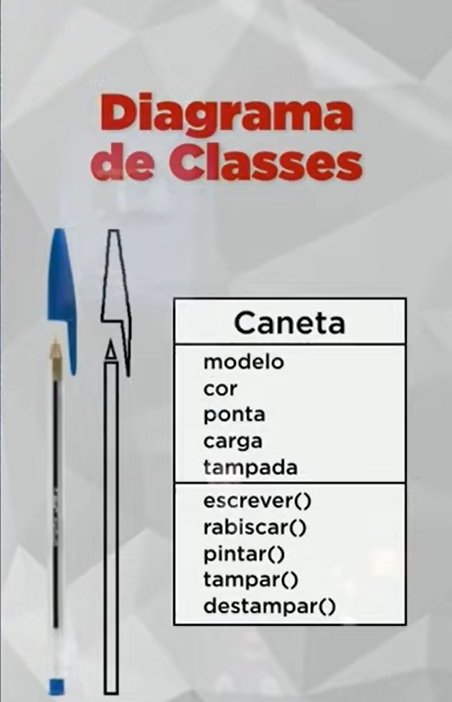
- O “this.” Chama o objeto que será referenciado.

**- UML (Linguagem de Modelagem Unificada):**

**- Diagrama de classes e Visibilidade:**

- Nos diagramas de classe, toda representação de classe é em forma de retângulo e este retângulo é dividido em três partes, sendo a primeira parte com o nome da classe iniciada com letra maiúscula, a segunda parte com os atributos iniciados com letras minúsculas e a terceira parte com os métodos, também iniciados com letras minúsculas, no entanto, no final possuem parênteses.

**- Por exemplo:**

****



- **Métodos Assessores:**  
- Getters: São métodos que pegam alguma coisa dentro do objeto.

- **Métodos Modificares:**  
- Setters: São métodos que modificam coisas que estão dentro do objeto.

-**Método Construtor:**  
- É um tipo especial de método que é chamado automaticamente quando um objeto de uma classe é criado. Sua principal função é inicializar o estado inicial do objeto, definindo os valores iniciais dos seus campos ou executando outras operações necessárias para preparar o objeto para uso.

**ATIVIDADE:**  
- O atributo “tipo” deve ser ou conta poupança(CP) ou conta corrente(CC);

- No método “abrirConta”, na hora de abrir a conta é pra mudar o status para verdadeiro, logo se a conta está fechada o status é falso, além disso, na hora de abrir a conta deve ser informado qual o tipo da conta CC ou CP, se for uma conta CC a pessoa já ganha R$50,00 na conta, se for CP a pessoa ganha R$150,00;

- No método “fecharConta” a pessoa não pode fechar a conta se a mesma ainda tiver saldo e se estiver devendo dinheiro na conta;

- No método “depositar” para a pessoa depositar a conta deve estar aberta, então o status deve estar verdadeiro;

- No método “sacar” para a pessoa sacar a conta deve estar aberta, então o status deve estar verdadeiro e deve haver saldo na conta (só pode ser sacado, no máximo, a quantidade de dinheiro que há na conta);

- No método “pagarMensalidade” não tem nada demais, porém, toda vez que esse método for chamado, o cliente que tiver CC deverá pagar o valor de 12 reais (será descontado do saldo da conta), Já o cliente que tiver CP deverá pagar o valor de vinte reais;

- No método construtor, sempre que uma nova conta for criada o seu status vai ser definido como falso, ou seja, a conta vai estar fechada e o saldo definido como zero.

**- Exemplo de método construtor:**

public ContaBanco(){

        saldo **=** 0;

        status **=** false;

    }

**- Exemplo de método comum:**

public ***void*** abrirConta() {

    }

- **Exemplo de métodos Getters e Setters:**

public ***float*** getNumConta() {

**return** this.numConta;

    }

    public ***void*** setNunConta(***int*** *n*) {

        this.numConta **=** n;

    }

- **Pilares da POO:**

- **Abstração** (Lembrar de facilitar/abstrair);

- **Encapsulamento** (Lembrar de capsula);

- A grosso modo: Serve pra proteger e criar um padrão.

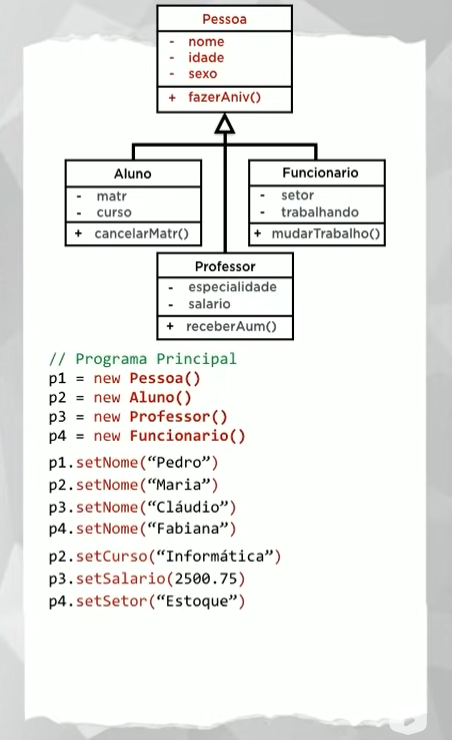
- Conceito: Encapsular é ocultar partes independentes da implementação, permitindo construir partes invisíveis ao mundo exterior.

PS.: Encapsular não é obrigatório, mas é uma boa prática.

- **Herança** (Lembrar de herança deixada por alguém);

- **Polimorfismo** (Lembrar – Várias formas de fazer a mesma coisa);

**-** **Representação de herança:**

****

