

# Programação Orientada a Objetos

Matheus Barbosa  
matheus.barbosa@dcx.ufpb.br

Como era a  
programação  
antigamente?

POO

**Programação Orientada a Objetos (POO):** Baseia-se em objetos que combinam dados e comportamentos, usando conceitos como classes, herança e encapsulamento.

Modular

**Programação Modular:** Divide o programa em módulos ou funções independentes, promovendo reutilização e organização do código.

Estruturada

**Programação Estruturada:** Organiza o código em blocos com uso de estruturas como *if*, *while* e *for*, facilitando a leitura e manutenção.

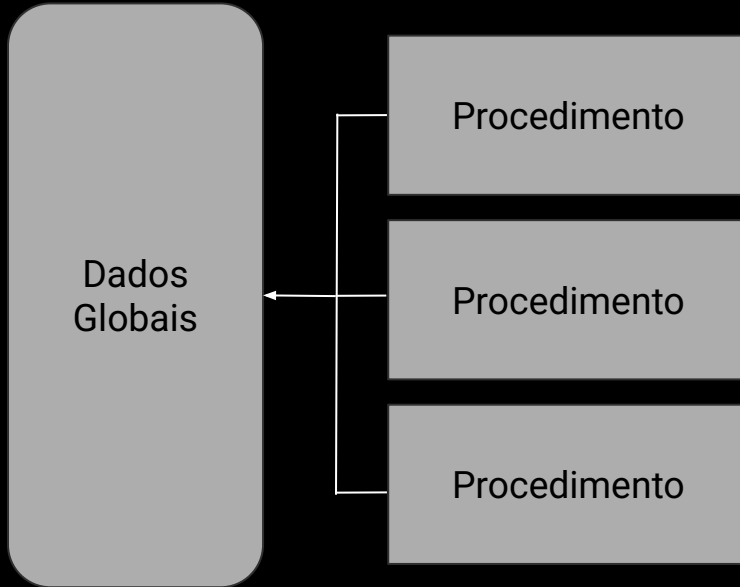
Linear

**Programação Linear:** Executa instruções em sequência, do início ao fim, sem desvios ou repetições. Ideal para programas simples.

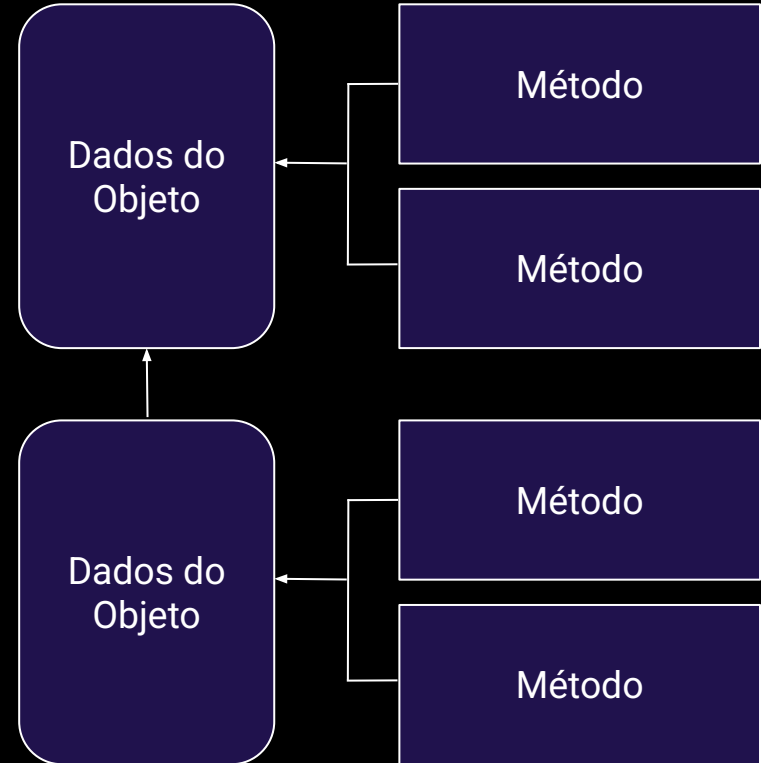
Baixo nível

**Linguagens de baixo nível:** formas bem simples de falar com o computador. Cada tipo de computador tinha sua própria linguagem, então o que funcionava em um, não funcionava em outro

## Como era?



## Como ficou?



- É preciso validar CPF

```
cpf = formulario->campo_cpf  
valida(cpf)
```

- O cliente precisa ter mais de 18 anos

```
idade = formulario->campo_idade  
valida(idade)
```



A responsabilidade de fazer verificações ficou espalhada por todo o seu código

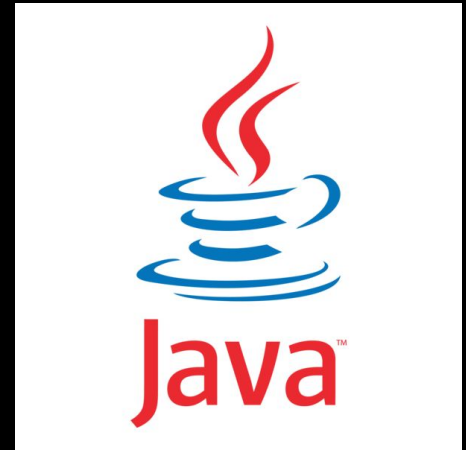












Seria interessante poder concentrar essa responsabilidade em um lugar só.



Mudar essa validação e os outros programadores nem precisassem ficar sabendo disso.

# Linguagens OO



May 2025	May 2024	Change	Programming Language		Ratings	Change
1	1			Python	25.35%	+9.02%
2	3	▲		C++	9.94%	+0.41%
3	2	▼		C	9.71%	-0.27%
4	4			Java	9.31%	+0.62%
5	5			C#	4.22%	-2.27%
6	6			JavaScript	3.68%	+0.66%
7	8	▲		Go	2.70%	+1.10%
8	7	▼		Visual Basic	2.62%	+0.61%
9	11	▲		Delphi/Object Pascal	2.29%	+1.05%
10	9	▼		SQL	1.90%	+0.45%

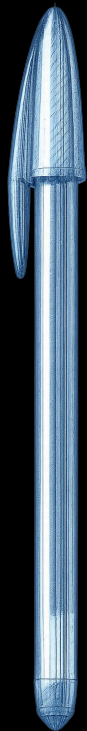
Fonte: <https://www.tiobe.com/tiobe-index/>

O que é um objeto?

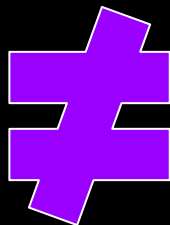




Um objeto é uma coisa **material** ou **abstrata** que pode ser percebida pelos sentidos e descrita por meio de duas **características**, **comportamentos** e **estado atual**.



Todo objeto vem de  
uma classe.  
(molde usado para  
gerar o objeto)



Classe

Objeto

## Características

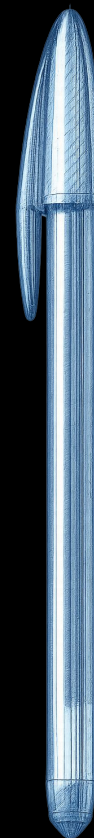
que coisas eu tenho?

## Comportamentos

que coisas eu faço?

## Estado atual

como eu estou agora?



## Características

que coisas eu tenho?

## Comportamentos

que coisas eu faço?

## Estado atual

como eu estou agora?

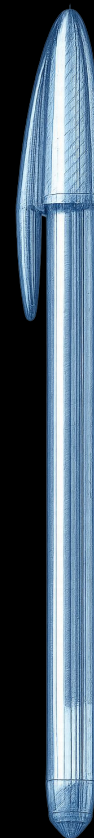
Modelo

Cor

Ponta

Carga

Tampada



## Características

que coisas eu tenho?

## Comportamentos

que coisas eu faço?

## Estado atual

como eu estou agora?

Modelo

Escrever

Cor

Rabiscar

Ponta

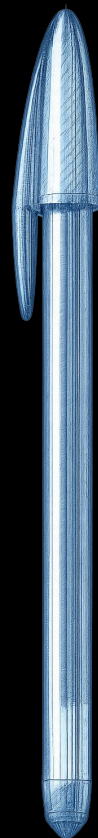
Pintar

Carga

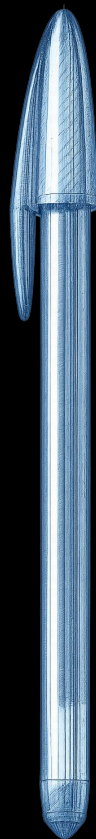
Tampar

Tampada

Destampar



Características que coisas eu tenho?	Comportamentos que coisas eu faço?	Estado atual como eu estou agora?
Modelo	Escrever	xxx-123
Cor	Rabiscar	Azul
Ponta	Pintar	0.5
Carga	Tampar	90%
Tampada	Destampar	Sim

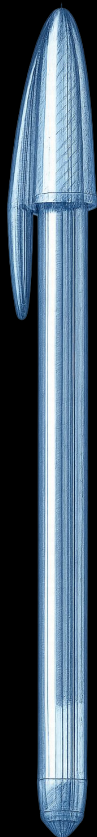


Características = Atributo  
que coisas eu tenho?

Comportamentos = Método  
que coisas eu faço?

Estado atual = Estado  
como eu estou agora?





## Classe Caneta

**modelo:** texto  
**cor:** texto  
**ponta:** decimal  
**carga:** inteiro  
**tampada:** logico

**rabiscar()**  
**escrever()**  
**tampar()**  
**destampar()**

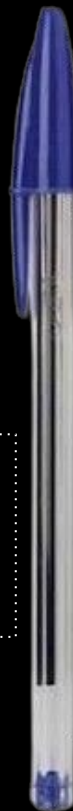
FimClasse

atributos

instanciar

```
c1 = nova Caneta()  
c1.modelo = xxx-123  
c1.cor = azul  
...
```

métodos



**modelo:** xxx-123  
**cor:** azul  
**ponta:** 0.5  
**carga:** 90%  
**tampada:** sim

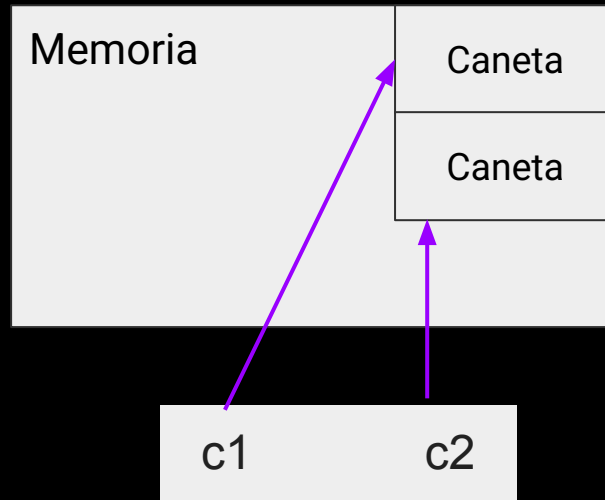
estado

```
c1 = nova Caneta()  
c1.modelo = xxx-123  
c1.cor = azul  
...  
  
c2 = nova Caneta()  
c2.modelo = yyy-123  
c2.cor = preta  
...
```



# Objetos são acessados por referências

```
c1 = nova Caneta()  
c1.modelo = xxx-123  
c1.cor = azul  
...  
  
c2 = nova Caneta()  
c2.modelo = yyy-123  
c2.cor = preta  
...
```



```
c1 = nova Caneta()  
c1.modelo = xxx-123  
c1.cor = azul  
...  
  
c2 = nova Caneta()  
c2.modelo = yyy-123  
c2.cor = preta  
...  
c2.destampar()
```



# Classe

Define os atributos e métodos comuns que serão compartilhados por um objeto

# Objeto

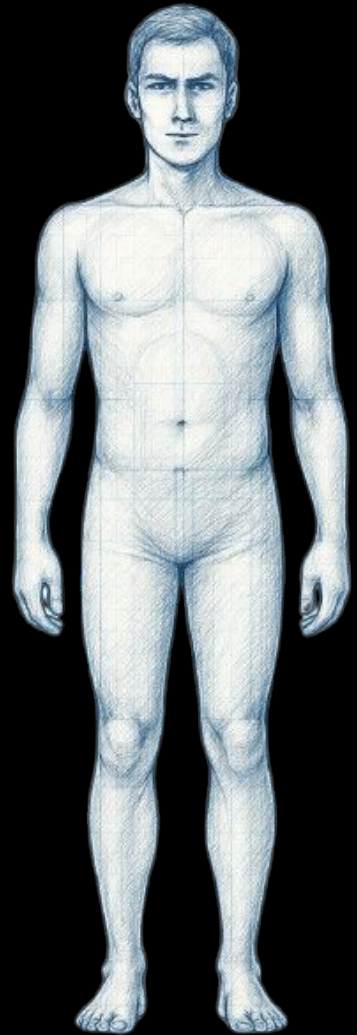
É a instância de uma classe

# Quais atributos e métodos uma pessoa pode ter?

Peso, altura, cor dos olhos...

## Precisamos listar todos?

Não. Assim como o desenho da pessoa abstrai características não essenciais para destacar a estrutura geral, a abstração em POO permite representar entidades do mundo real de forma simples, clara e funcional, capturando **apenas o necessário** para o problema que se quer resolver.



**Abstração** é o processo de simplificar a **complexidade** do mundo real ao modelar objetos, focando apenas nos aspectos essenciais e **ignorando os detalhes irrelevantes** para o **contexto atual**.

## Atividade

1. Identifique um objeto **físico** do seu ambiente atual e classifique-o
2. Identifique um objeto **conceitual** do seu dia-a-dia e classifique-o