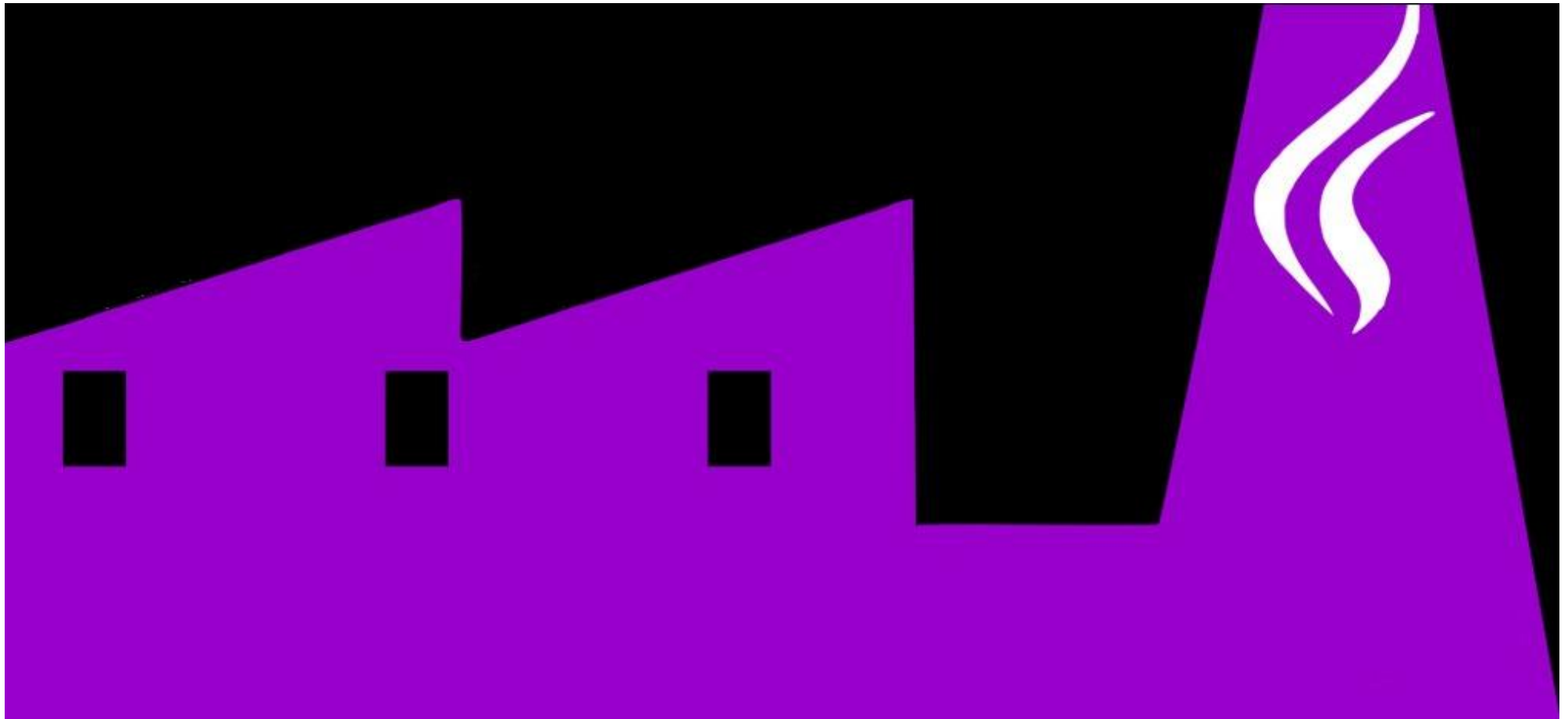


# Fábrica de Software

Prof<sup>es</sup>. Ivan L. Süptitz e Evandro Franzen

Introdução à Programação Orientada a Objetos



# O que é a sigla POO?

**POO** = **P**rogramação **O**rientada a **O**bjetos

**Definição:** É um **paradigma** de programação de computadores onde se usam **classes** e **objetos** criados a partir de **Modelos**, para **representar** e **processar dados**.

# O que é um paradigma?

- É um exemplo que serve como modelo. Um padrão. Uma forma de pensar;



- Colombo mudou o paradigma da navegação

# Quais pilares sustentam a POO?

# POO

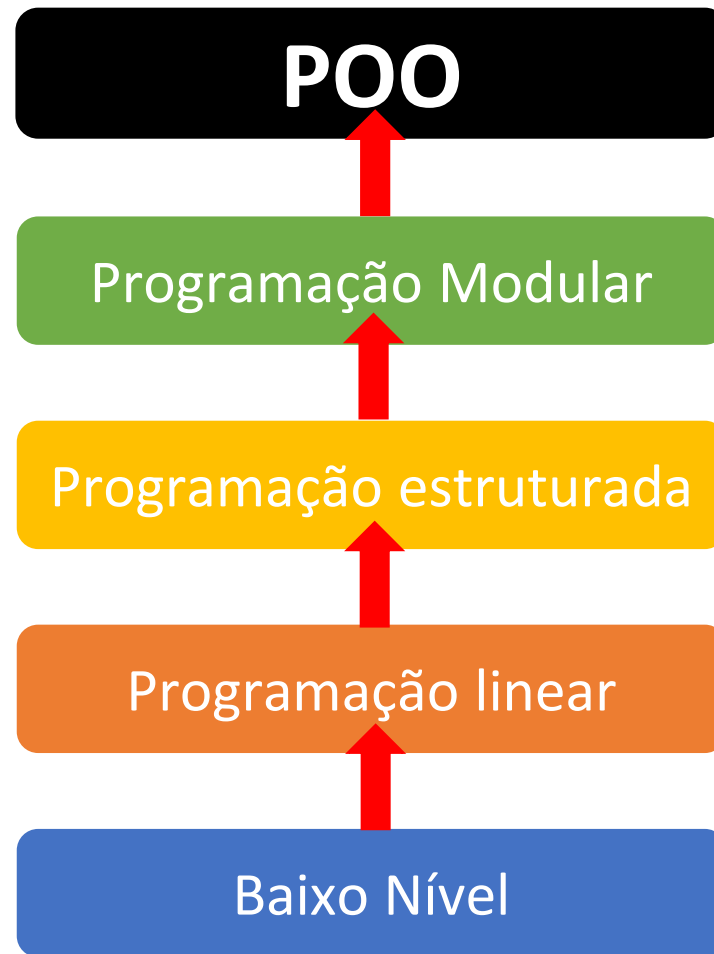
---



# Objetivo principal de POO?

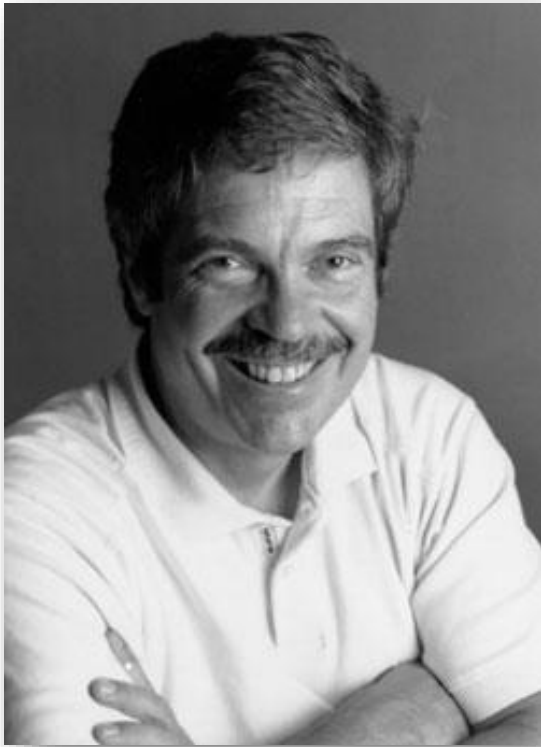
Aproximar o mundo real do mundo digital (facilitando a compreensão dos objetos)

# Antigamente, como eram as programações?



# Quem criou (pai da) POO?

- Quando: 1970
- Formado em: Matemática e Biologia
- Onde trabalhou: Xerox (smalltalk), Atari, Apple, Walt Disney, HP (mais conhecidos).



**Alan Kay**

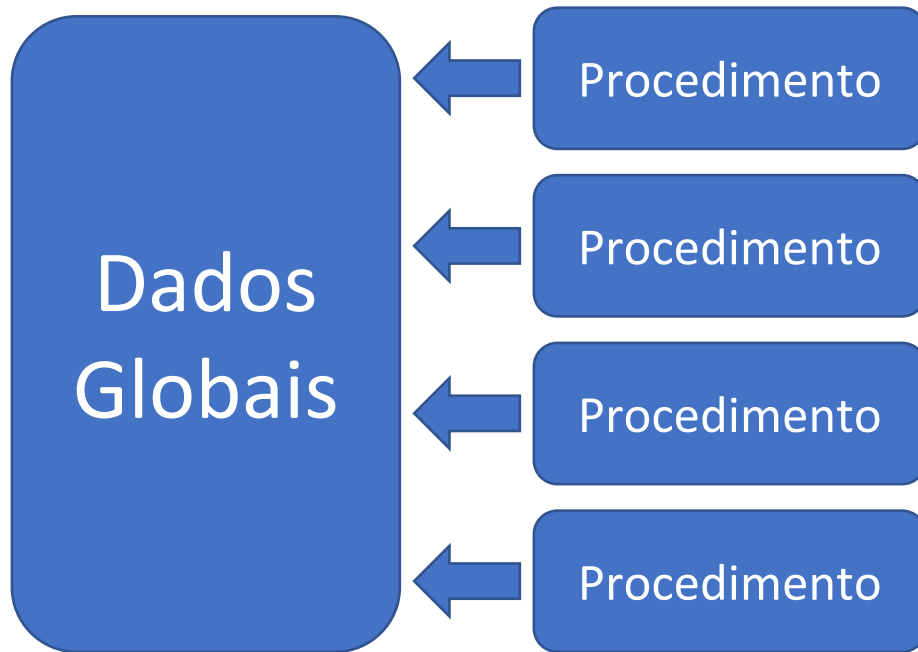


**Premio Turing**

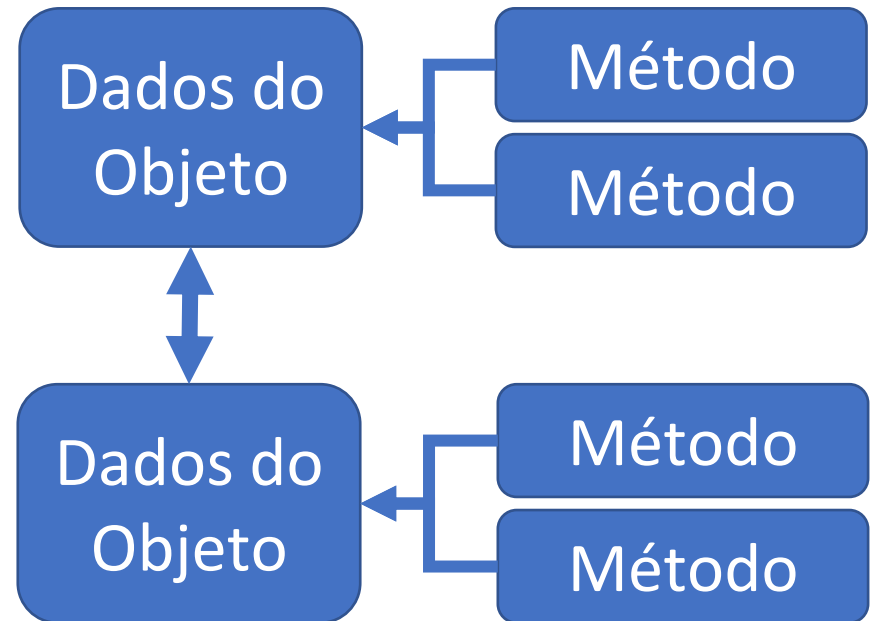
**Postulado:** “ ...de que o computador ideal deveria funcionar como um organismo vivo, isto é, cada célula comportar-se-ia relacionando-se com outras a fim de alcançar um objetivo, funcionando de forma autônoma. As células poderiam também reagrupar-se para resolver um outro problema ou desempenhar outras funções.”

# Comparando...

## Como era?



## Como ficou?





# Exemplos de LP00?



JavaScript



Objective-C



# Vantagens da POO?

- Confiável (Isolamento)
- Oportuno (Desenvolvimento paralelo)
- Fácil Manutenção (Beneficiar todos os objetos)
- Extensível (Permite crescimento)
- Reutilizável (Outros sistemas)
- Natural (Fácil de entender como funciona; menos detalhes de implementação).



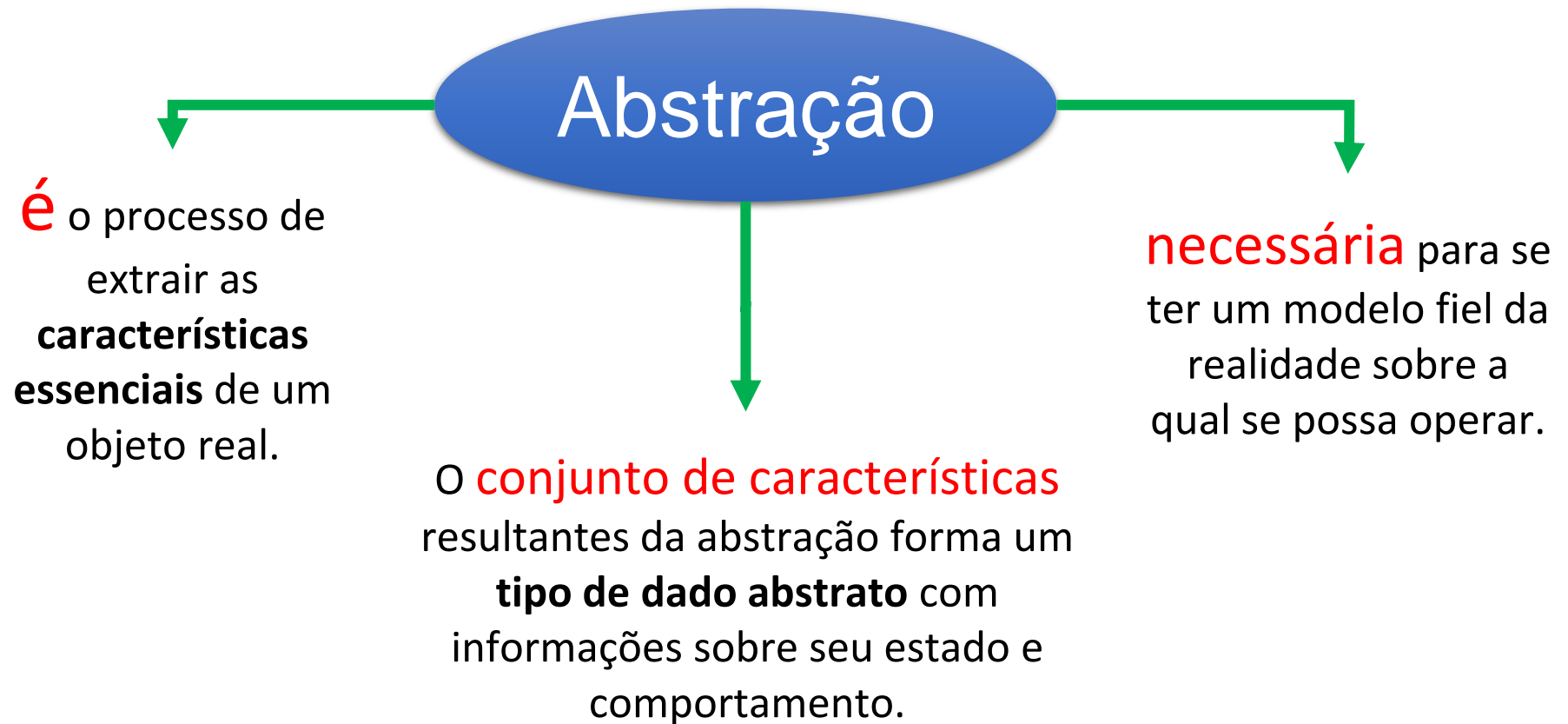
# Dúvidas até o momento?

- Vamos nos aprofundar nos conceitos de POO.
- Já ouviram falar em:
  - Objeto?
  - Classe?
  - Atributo?
  - Método?

**Este é mais um objetivo da aula de hoje!!!**

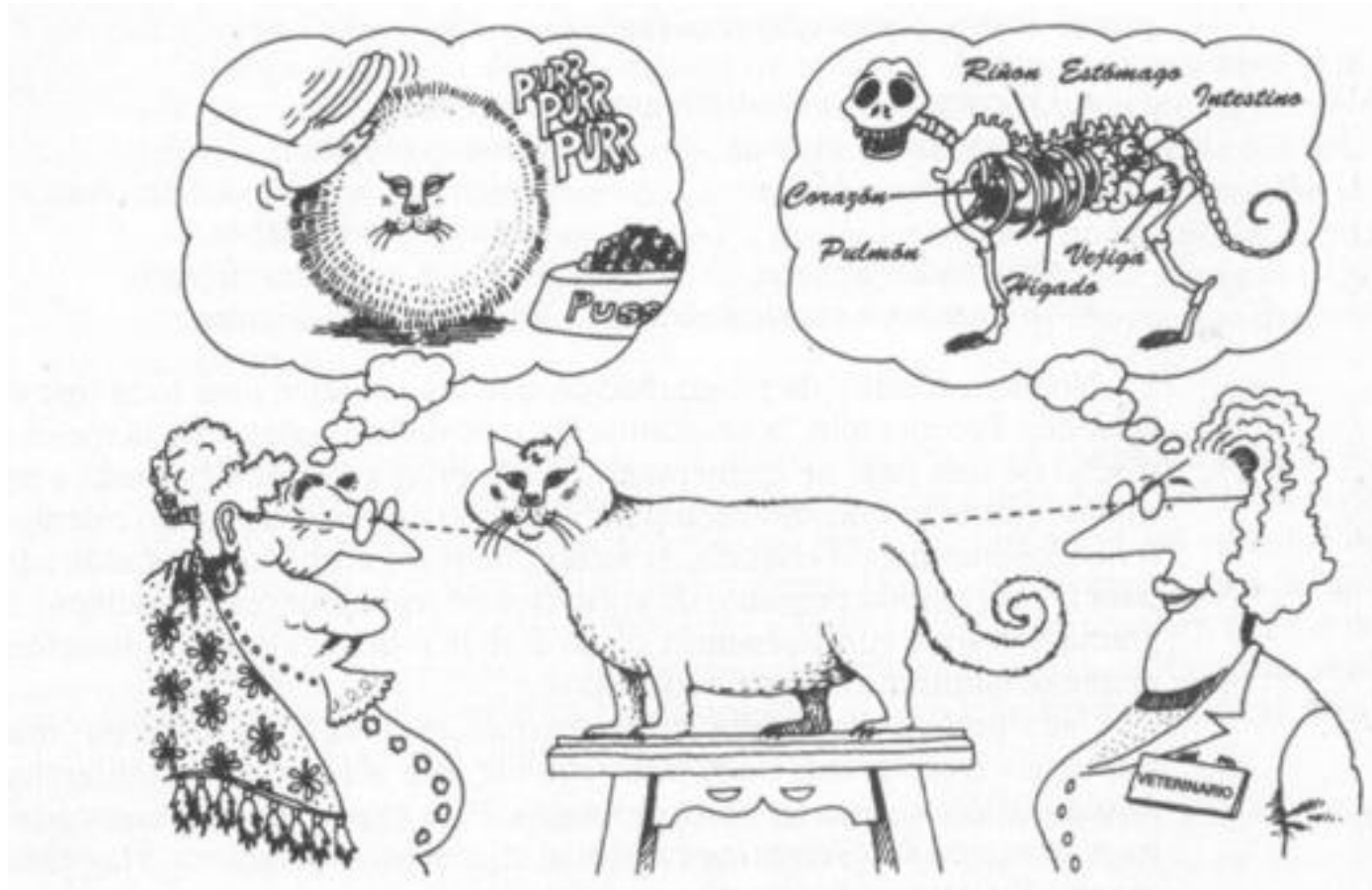
**Saber identificar, diferenciar e construir exemplos.**

# Retomando – O que era abstração?



# Retomando – O que era abstração?

Ex1.



Abstração é o processo de extrair as características essenciais de um objeto real.

A abstração é necessária para se ter um modelo fiel da realidade sobre a qual se possa operar.

# Mundo Real – Objetos reais



**Dados** (características):  
cor, marca, modelo, ano ...  
**Instruções** (ações): trocar  
de marcha, mover,  
consertar, ligar, trocar  
pneus ...



**Dados** (características):  
sabor, peso, tipo ...  
**Instruções** (ações):  
comer, descascar, lavar ...



**Dados** (características):  
tamanho, cor, temperatura ...  
**Instruções** (ações): adicionar  
açúcar, mexer, beber ...

# Mundo Virtual – Objetos abstratos



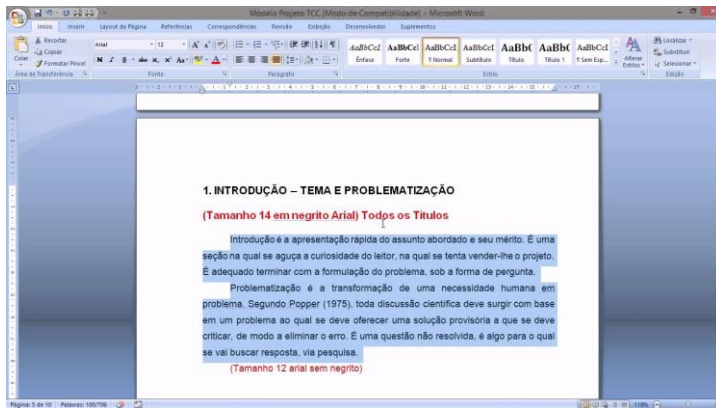
**Dados** (características):  
tamanho, localização,  
título, cor de fundo ...

**Instruções** (ações): abrir,  
fechar, mover, expandir ...



**Dados**  
(características):  
dia, mês, ano...

**Instruções**  
(ações): validar,  
inicializar,  
exibir, agendar,  
...



**Dados** (características):  
espaçamento, margens,  
alinhamento, ...

**Instruções** (ações): escrever,  
copiar, mover, ...

# O que é um objeto em POO?

- Ex:

- Celular ( ) Sim ( ) Não
- Carro ( ) Sim ( ) Não
- Ano de fabricação ( ) Sim ( ) Não
- Mouse ( ) Sim ( ) Não
- Agenda de reuniões ( ) Sim ( ) Não
- Disciplina de FS ( ) Sim ( ) Não
- Aprender OO ( ) Sim ( ) Não

**Definição:** coisa **material** ou **abstrata** que pode ser **percebida** pelos sentidos e **descrita** por meio de suas **características**, **comportamento** e **estado atual**.



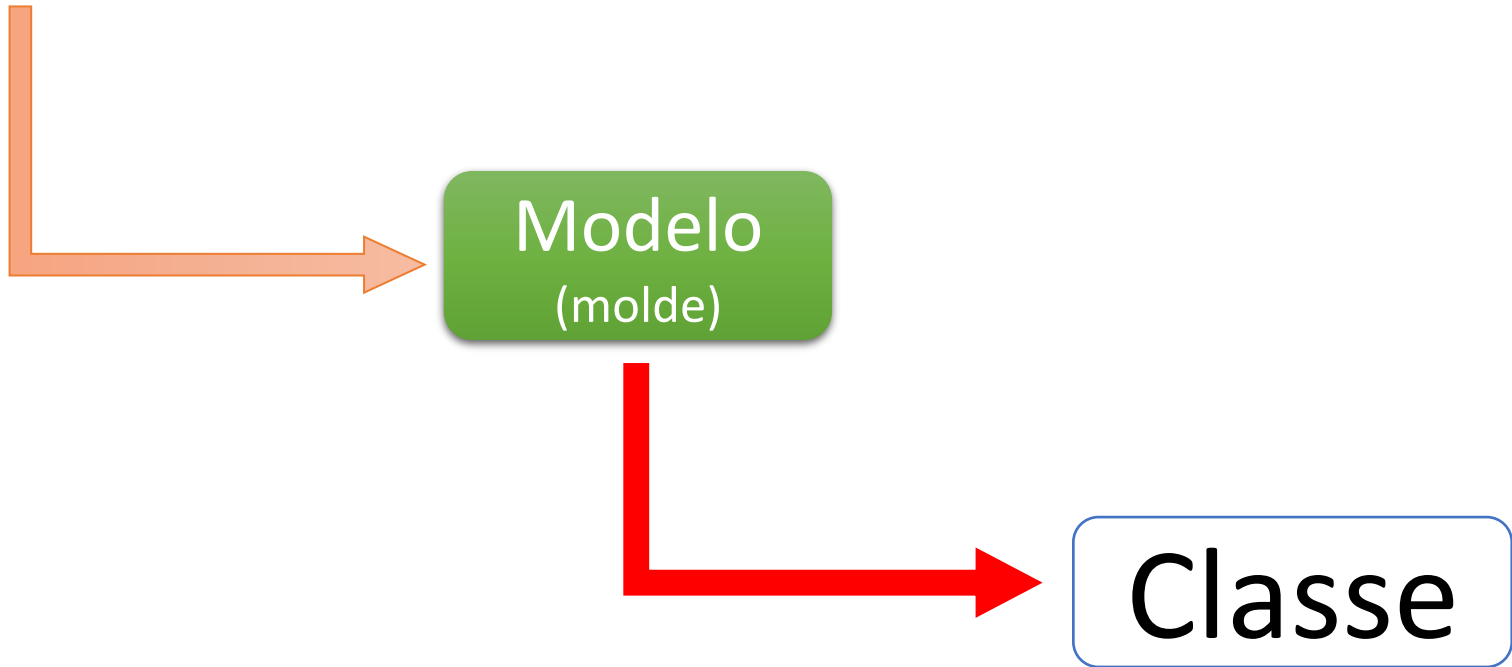
# O que é um objeto ...

- Outro exemplo:

- Caneta

- Material
    - Percebida pelos sentidos
    - Descrita

? O que eu preciso para produzir uma caneta?



# Perguntas sobre o que colocar na Classe (modelo)?

- **O que o objeto pode ter?**  
Características => **Atributos**
- **O que o objeto pode fazer?**  
Ações => **Métodos**
- **Como o objeto esta agora?**  
Estados => **Atributo.**



E da caneta?

1 – Características?

2 – Ações?

3 – Estados?

# Vamos exercitar, como criar a Classe Caneta?

Nome da Classe

```
1  classe Caneta
```

```
2      modelo : caractere  
3      cor    : caractere  
4      ponta  : real  
5      carga  : inteiro  
6      tampada : logico
```

→ **Atributos**

```
7  
8      metodo rabiscar()  
9          se (tampada) então  
10             escreva ("ERRO")  
11             senão  
12                 escreva ("Rabiscando!!!")  
13      fim Metodo
```

→ **Método  
1**

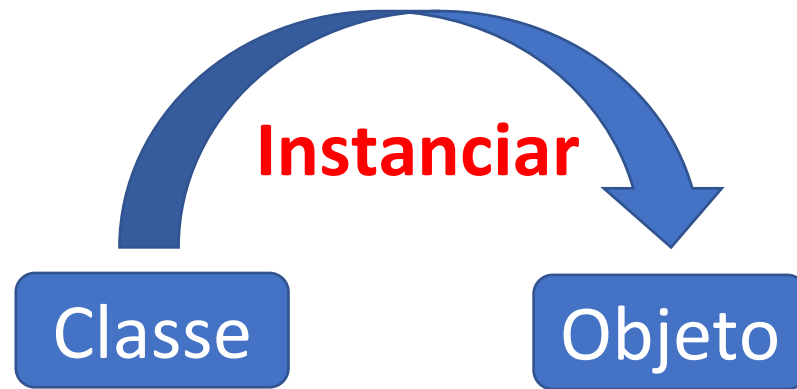
```
14  
15      metodo tampar()  
16          tampada : verdadeiro  
17      fim Metodo
```

→ **Método  
2**

```
18  
19  fim Classe
```

# E agora, como faço para gerar um objeto?

- Preciso instanciar o objeto.
- Como assim?



# Como fica o meu algoritmo para instanciar uma caneta?

- 1 - Como alterar os atributos da c1?
- 2 – Como chamar um método?
- 3 - Consigo criar um segundo objeto com a mesma classe? Como?

**Instanciar**

**Nome do objeto**      **Classe**

```
1  c1 = nova Caneta
2
3  c1.cor = "azul"
4  1 c1.ponta = 0.5
5  c1.tampada = falso
6
7  c1.rabiscar() 2
8
9  c2 = nova Caneta
10 3 c2.cor = "vermelha"
```

# Resumindo...

- **Classe:** modelo que define os atributos e métodos comuns que serão compartilhados por um objeto.
- **Objeto:** é a instância de uma classe.

# Exercício J.2.1

- Identificar dois objetos concretos e um objeto abstrato de seu ambiente (trabalho ou casa) e classifique-os (criando sua classe, seus atributos e métodos).
- Criar três objetos com as classes da atividade anterior. Depois, modificar três atributos de cada objeto. Também, realizar a chamada de dois métodos para cada objeto.

# Exercício J.2.2

Reescreva o programa  
(estruturado em C) a seguir  
de acordo com o paradigma  
de Programação Orientada a  
Objetos (em Java)

```
typedef struct Aluno{  
    int matricula;  
    char nome[50];  
    float n_p1;  
    float n_p2;  
    float n_trab;  
    float n_ex;  
    float n_media;  
}Aluno;
```

```
float calcula_media(float n1, float n2, float n3, float n4){  
    return (n1+n2+n3+n4)/4.0;  
}  
  
void main(){  
    Aluno a;  
    scanf("%f %f %f %f", &a.n_p1, &a.n_p2, &a.n_trab, &a.n_ex);  
    a.n_media = calcula_media(a.n_p1, a.n_p2, a.n_trab, a.n_ex);  
}
```



# Dicas

- Como aprender programação?
  - Estude outros códigos-fonte
  - Faça todos os exercícios
    - Do início até o final
- A universidade é uma '*simulação*' do ambiente empresarial ou acadêmico;
- Estudamos **fundamentos** e exercitamos a capacidade de **resolver** quaisquer problemas do mundo real.

# Referências

- Documentação oficial Java:  
<https://docs.oracle.com/javase/tutorial/java/index.html>
- DEITEL, Paul J.; DEITEL, Harvey M. Java: como programar