

# CONCEITOS DE ORIENTAÇÃO A OBJETO COM PYTHON

Prof. Emmerson Santa Rita da Silva.

00

# PROGRAMAÇÃO ORIENTADA A OBJETO

Python



Digital Transformation  
Z Academy

# Ementa

- Conceitos do paradigma da programação orientada a objetos
- Abstração
- Introdução a classes e objetos
- Agregação e Composição de objetos
- Encapsulamento, herança, polimorfismo
- Projeto orientado a objetos
- Linguagem de programação orientada a objetos

# Objetivo Geral

- Tornar o aluno apto a **compreender** os **conceitos** de programação orientada a objetos e **projetar**, **desenvolver** e **testar** softwares utilizando o paradigma e uma linguagem de programação orientado a objetos.

# Objetivo s Específicos

- Compreender os conceitos básicos do paradigma de programação orientado a objetos
- Utilizar classes, objetos e interação entre objetos.
- Entender conceitos como abstração, encapsulamento, herança e polimorfismo.
- Projetar software orientado a objetos
- Desenvolver software utilizando uma linguagem de programação orientada a objetos.
- Testar softwares desenvolvidos no paradigma de orientação a objetos.

# VIDA DE PROGRAMADOR

.COM.BR

/" HISTÓRIA REAL  
ENVIADA POR  
WAGNER "/



#344

CARA... O QUE É  
"ORIENTADO A OBJETOS"?



É UMA FORMA DE PROGRAMAR  
QUE FOI CRIADA PARA TENTAR  
APROXIMAR O MUNDO REAL DO  
MUNDO VIRTUAL USANDO  
PROPRIEDADES, OBJETOS,  
MÉTODOS, ETC...

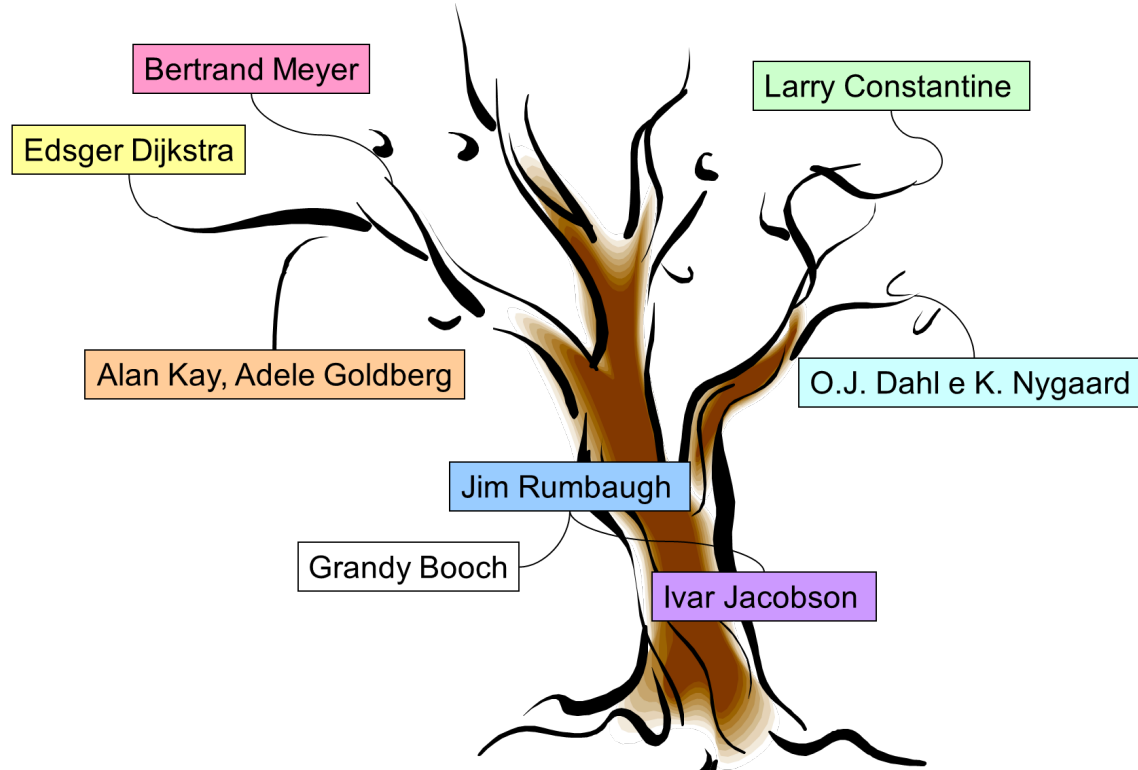


AHHHH TÁ... É QUE EU  
ACHO QUE FAÇO ORIENTADO  
A OBJETOS E NEM SEI...



# 01

## Um breve Histórico



# Paradigma Orientado a Objetos

- Estamos rodeados por objetos: mesa, carro, livro, pessoa, etc; e Os objetos do mundo real têm duas características em comum:
  - **Estado = propriedades** (nome, peso, altura, cor, etc.);
  - **Comportamento = ações** (andar, falar, calcular, etc.).
- Paradigma para desenvolvimento de software que baseia-se na **utilização de componentes individuais (objetos)** que colaboram para construir sistemas mais complexos.
  - A colaboração entre os objetos é feita através do envio de mensagens.
  - Um paradigma é um conjunto de regras que estabelecem fronteiras e descrevem como resolver problemas dentro desta fronteira.



# Paradigma Orientado a Objetos

- **VANTAGENS**

- Os modelos refletem o mundo real de maneira mais aproximada:
  - Descrevem de maneira mais precisa os dados;
  - Mais fáceis de entender e manter.
- Pequenas mudanças nos requisitos não implicam em grandes alterações no sistema em desenvolvimento.

# Os quatro pilares

**ABSTRAÇÃO**

**ENCAPSULAMENTO**

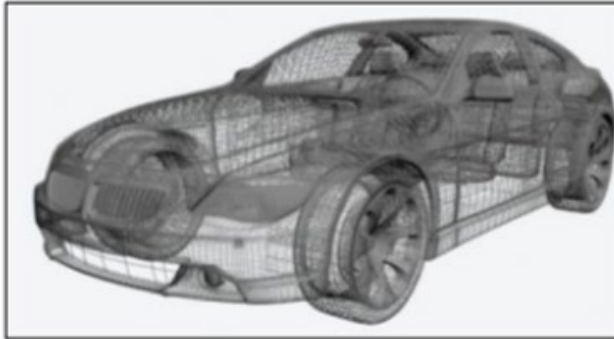
**HERANÇA**

**POLIMORFISMO**

# Paradigma Orientado a Objetos

- **CLASSES**

- A estrutura fundamental para definir novos objetos;
- Uma classe é definida em código-fonte. :



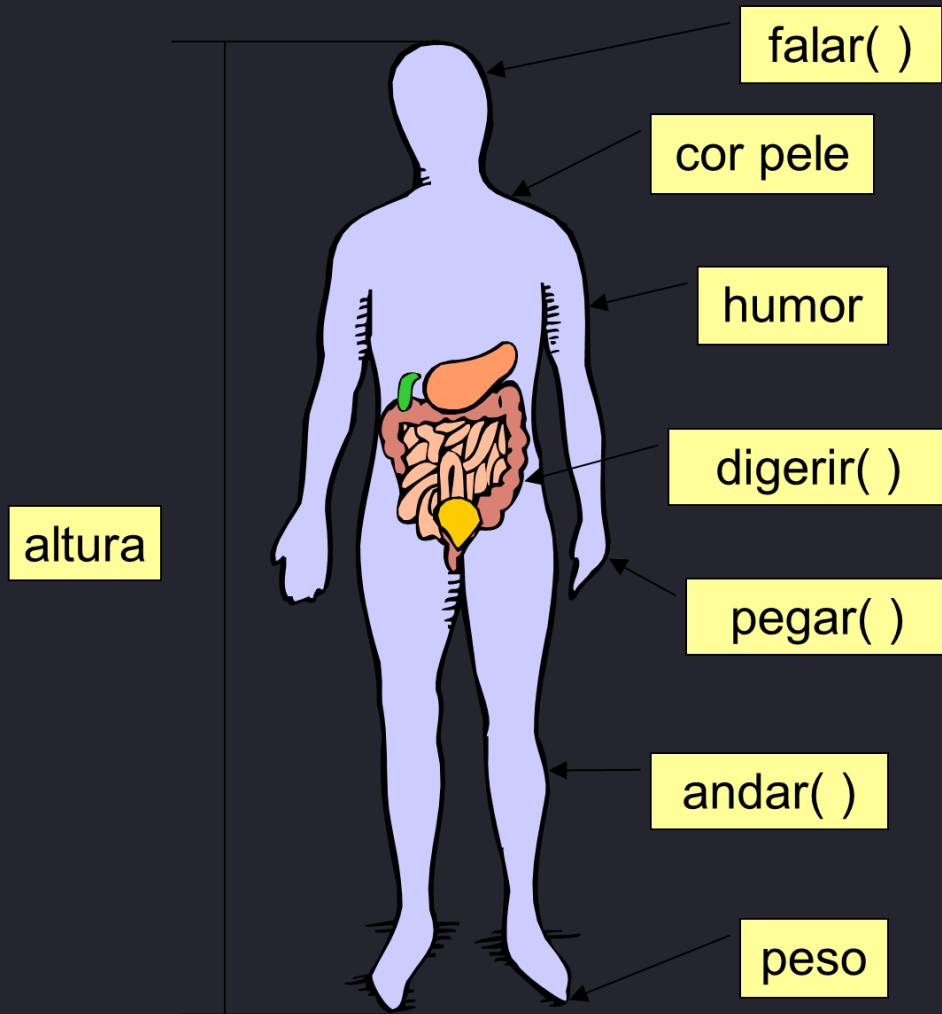
**Classe**



**Objeto**

# Conceitos preliminares

- **Objetos** são entidades que possuem características (**atributos** ) e comportamento (**operações** )



# Conceitos preliminares

- Atributo e métodos



Dividem-se em

Atributos

Operações

Carro
-Potência
-Cilindrada
-Acessórios
-Vel. Máxima
-Economia
+Andar()
+Frear()
+Manobrar()



Dividem-se em

Atributos

Operações

Ferramenta
-Propósito
-Tamanho
-Cor
-Material
+Girar()
+Bater()
+Encaixar()

# Conceitos preliminares

- Instanciação



Bicicleta da Ana

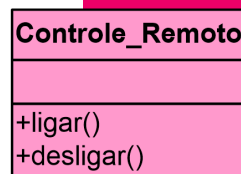
Bicicleta
-cor : char
-fabricante : char
-modelo : char
-no.serie : byte
-marcha : int
+trocar(in marcha)
+pedalar()



Bicicleta do Paulo

# Conceitos preliminares

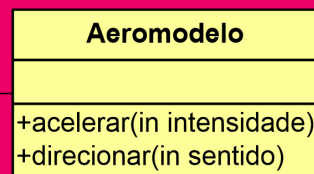
- Mensagens
- É o veículo pelo qual um objeto remetente obj1 transmite a um objeto obj2 um pedido para obj2 aplicar um de seus métodos



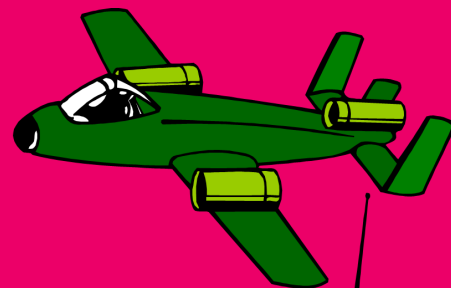
controla

1

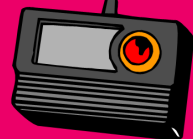
1



obj1



obj2



# Conceitos preliminares

- Mensagens





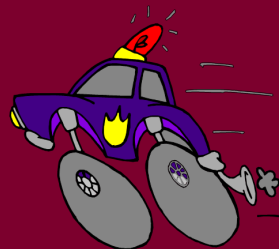
# Conceitos preliminares

- Mensagens (tipos)
- Informativa
- Interrogativa
- imperativa

**Informativa:** “funcionário se casou ontem, atualize data de casamento”.



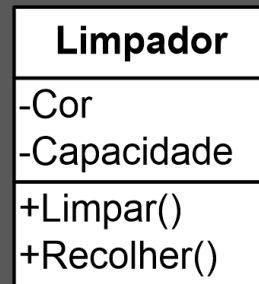
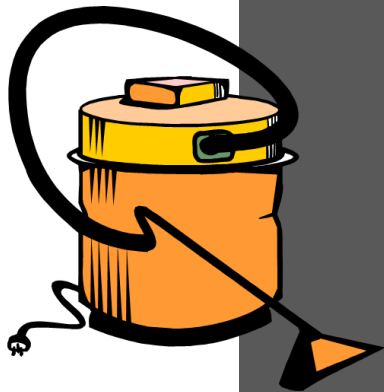
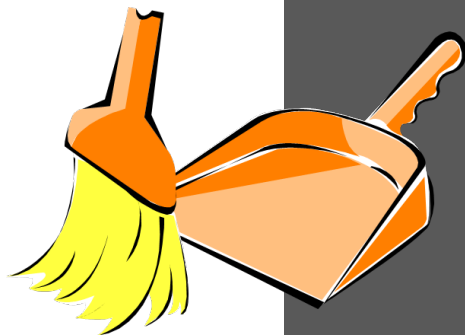
**Interrogativa:**  
“qual é sua  
posição  
atual?”



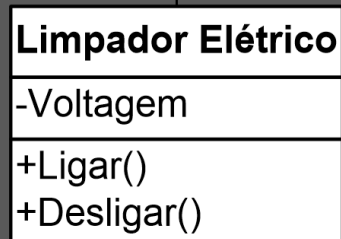
**Imperativa:** “acelerar velocidade”

# Conceitos preliminares

- Herança



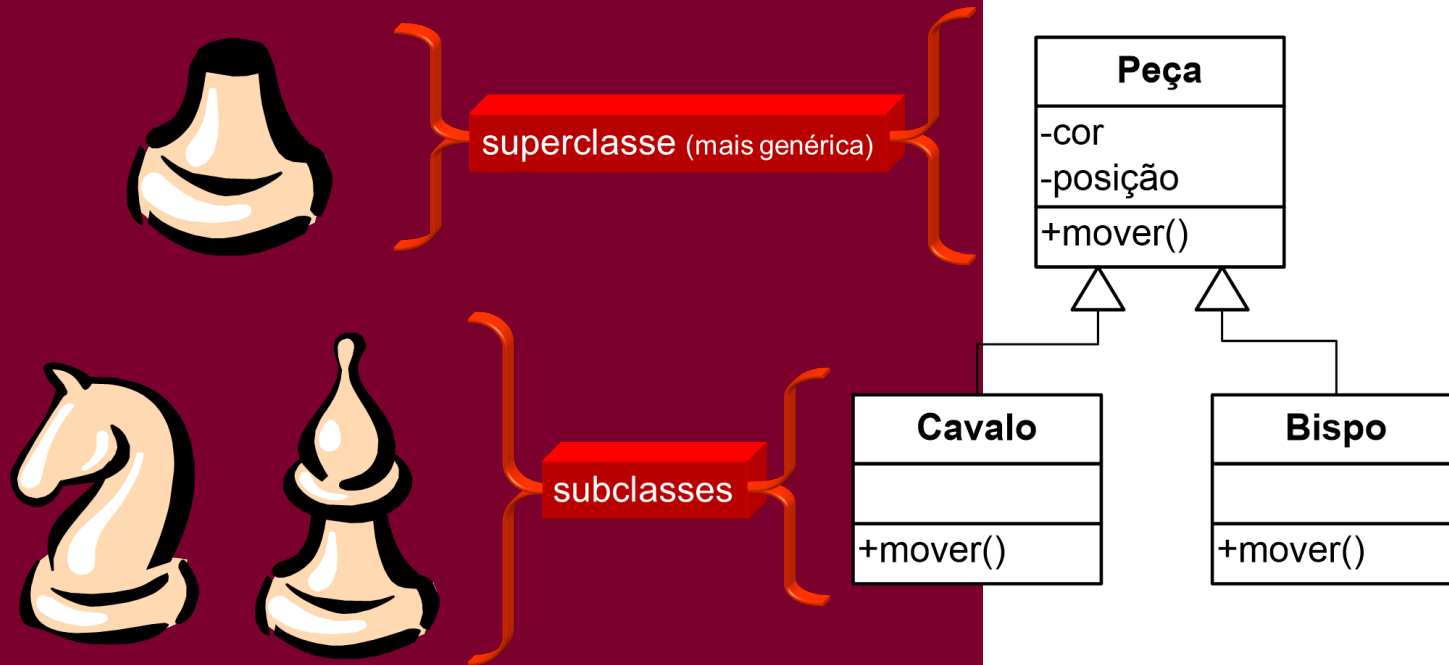
superclasse



subclasse

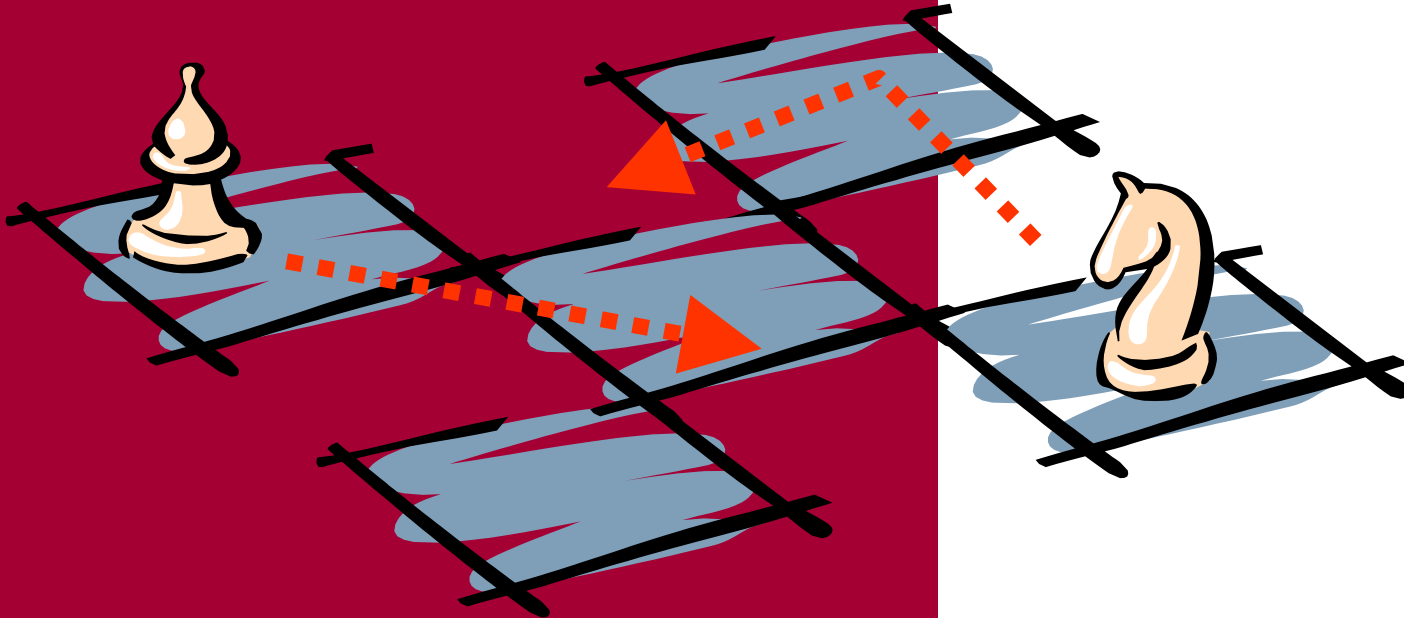
# Conceitos preliminares

- Polimorfismo



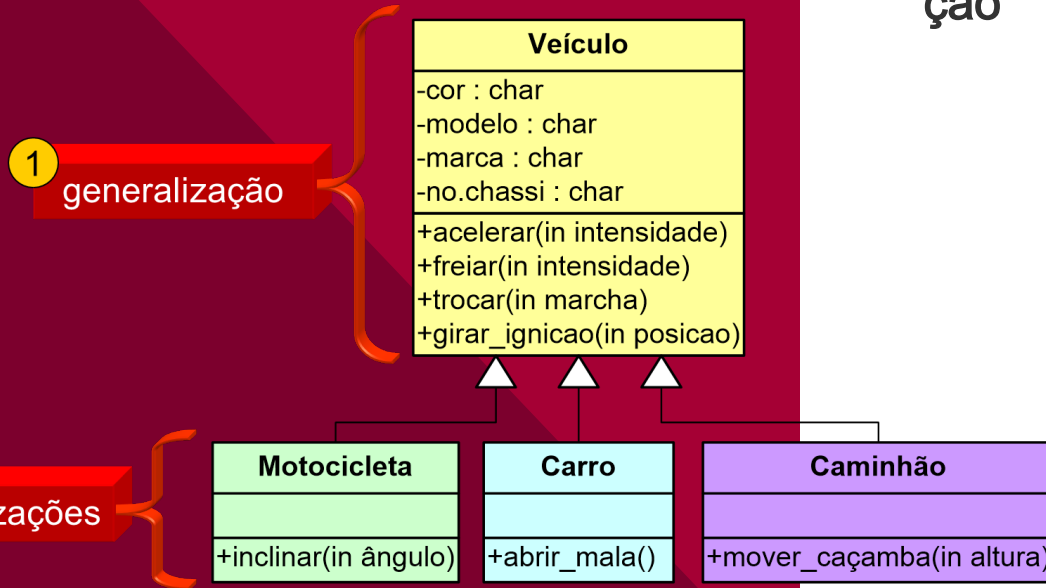
# Conceitos preliminares

- Polimorfismo



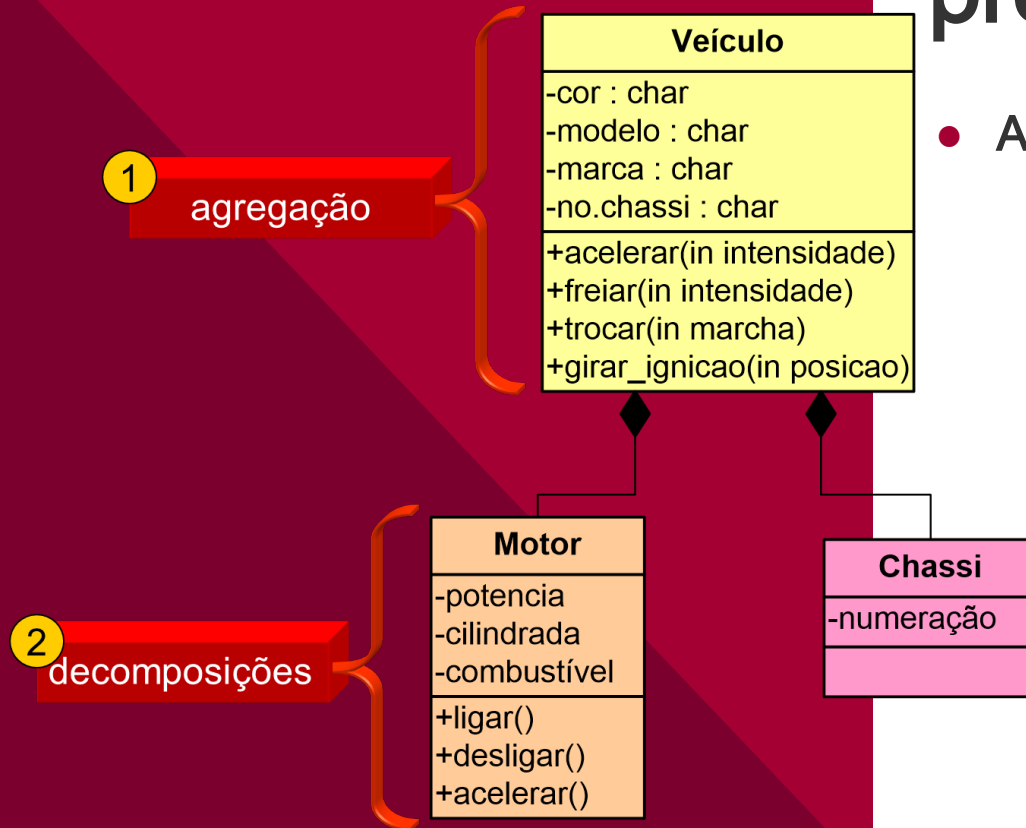
# Conceitos preliminares

- Generalização/Especialização



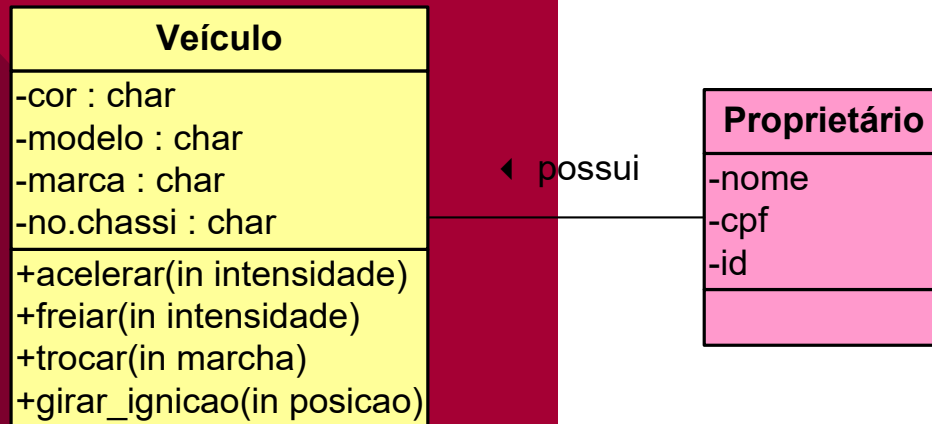
# Conceitos preliminares

- Agregação/Decomposição



# Conceitos preliminares

- Associação



# Conceitos preliminares

- Mais sobre atributos...





# Classes em Python

▶ Estrutura:

**class** nome\_da\_classe:

atributos

métodos

▶ Exemplo:

```
class Conta:  
    numero = 000000  
    saldo = 0.0
```



# INSTÂNCIA

- Uma instância é um objeto criado com base em uma classe definida;
- Classe é apenas uma estrutura, que especifica objetos, mas que não pode ser utilizada diretamente;
- Instância representa o objeto concretizado a partir de uma classe;
- Uma instância possui um ciclo de vida:



# Instância em Python

## ▶ Estrutura

variável = **Classe()**

## ▶ Exemplo

```
if __name__ == '__main__':  
    conta = Conta()  
    conta.saldo = 20  
    conta.numero = "13131-2"  
    print(conta.saldo)  
    print(conta.numero)
```

# MÉTODOS

- Representam os comportamentos de uma classe;
- Permitem acesso a atributos, tanto para recuperar os valores, como para alterá-los caso necessário;
- Podem retornar ou não algum valor; e
- Podem possuir ou não parâmetros.

# Métodos em Python

## ESTRUTURA:

def nome\_do\_metodo(self, parametros)

- Importante: o parâmetro **self** é obrigatório.

## EXEMPLO:

```
class Conta:
    numero = "00000-0"
    saldo = 0.0

    def deposito(self, valor):
        self.saldo += valor

    def saque(self, valor):
        if (self.saldo > 0):
            self.saldo -= valor
        else:
            print("Saldo insuficiente")

if __name__ == '__main__':
    conta = Conta()
    conta.saldo = 20
    conta.numero = "13131-2"
    print(conta.saldo)
    print(conta.numero)
```

# Método Contrutor

- Determina que ações devem ser executadas quando da criação de um objeto; e
- Pode possuir ou não parâmetros.

► Estrutura:

Def `__init__(self, parametros)`

► Exemplo:

```
class Conta:
    numero = "00000-0"
    saldo = 0.0

    def __init__(self, numero, saldoInicial):
        self.numero = numero
        self.saldo = saldoInicial

conta = Conta("12345-1", 0)
print(conta.numero)
print(conta.saldo)
```





# Atividade

## Calculadora de IMC em Python

Crie um programa em Python que calcule o Índice de Massa Corporal (IMC) com base no peso e altura fornecidos pelo usuário. Após calcular o IMC, o programa deve exibir uma mensagem indicando a faixa de peso da pessoa.

### Instruções:

1. Solicite ao usuário que insira seu peso (em quilogramas) e sua altura (em metros).
1. Calcule o IMC usando a fórmula:  $IMC = peso / (altura * altura)$ .
1. Exiba uma mensagem indicando a faixa de peso da pessoa com base no IMC calculado.



# Atividade

- **Classe Pessoa:** Crie uma classe que modele uma pessoa:
  - **Atributos:** nome, idade, peso e altura
  - **Métodos:** Envelhecer, engordar, emagrecer, crescer.
  - **Obs:** Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm
  - Pode possuir ou não parâmetros.





# Referências

1. DEITEL, H.M. Java, como programar. Porto Alegre: Bookman, 2005.
2. CORMEN, T.H. Algoritmos: Teoria e Prática. Campus, 2002.
3. LEMAY, Laura. Aprenda em 21 dias JAVA 2. Ed. Campus, 2003.
4. FURGERI, Sergio. JAVA 2: ENSINO DIDÁTICO. Ed. Erica, 2002.
5. KÖLLING, M. Programação Orientada a Objetos com Java: uma introdução prática usando o BLUEJ. São paulo : Pearson Prentice hall, 2004.
6. G BOOCH.Object -oriented analysis and design with applications . Redwood City, Calif .: Benjamin Cummings Pub. Co., 1994.
7. B J COX, A J NOVOBILSKI.Object -oriented programming . Reading: Addison Wesley, 1991.
8. T BUDD.Introduction to object -oriented programming . Reading, Mass.: Addison -Wesley Pub. Co., 1991.
9. D KAFURA.Object -oriented software design and construction with Java. Prentice Hall, 1999.

# OBRIGADO!

Prof.Emmerson Santa Rita da Silva.

✉ Emmerson.silva@ifam.edu.br

📷 @emmerson2013



Digital Transformation  
**Z Academy**



Digital Transformation  
**Z Academy**



**LG**

**incva**  
POLO DE INOVAÇÃO IFAM



**INSTITUTO FEDERAL**  
Amazonas  
Campus Manaus Zona Leste

