

# Introdução a Programação Orientação a Objetos I

Código da turma no classroom: iqmtxhd

Aulas: Segunda - 20:20 e Quinta - 18:30

# Para aprofundar!

Acessem os links na caixinha.

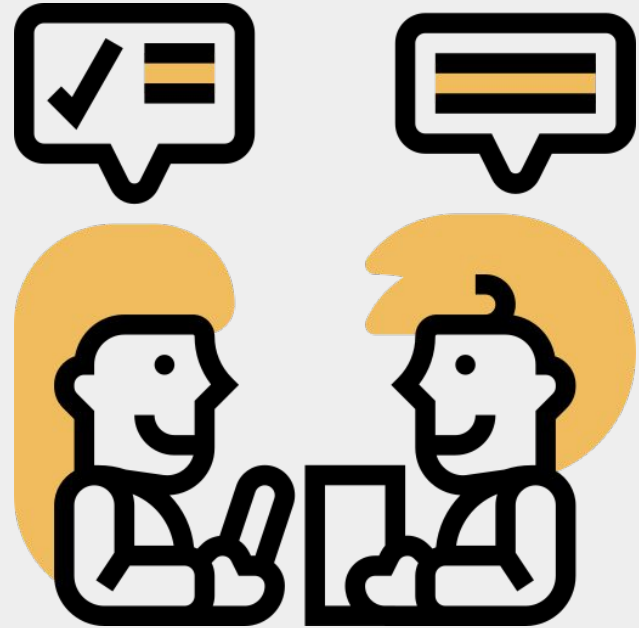
Acessar a  
documentação do  
Github: ver  
aspectos gerais,  
recursos e  
instruções iniciais

Solicitar **GitHub**  
**Student**  
**Developer Pack**

Série de vídeos  
sobre git e  
github

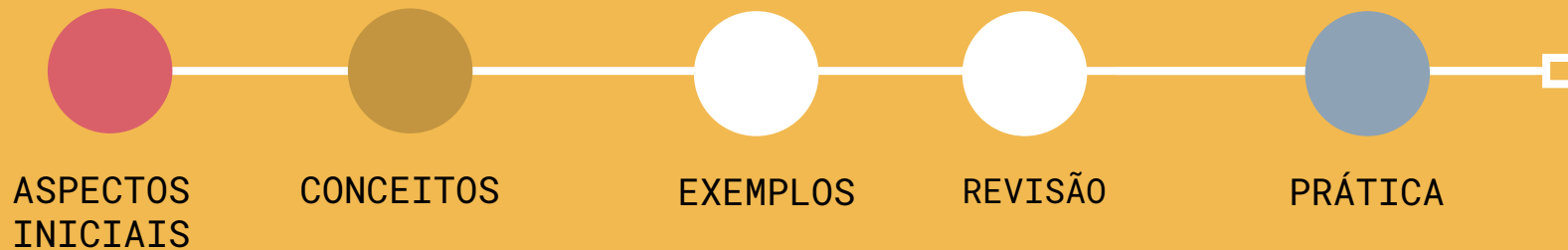
# Projeto Prático

- **Primeiro passo:** criar o repositório github do seu projeto e adicionar uma breve descrição da sua ideia.
- **Enviar:** enviem o link via classroom.



# Conceitos de Orientação a Objetos

# O QUE TEREMOS NA AULA DE HOJE!



# Programação Orientação a Objetos

A orientação a objetos, também conhecida como **Programação Orientada a Objetos (POO)** ou **Object-Oriented Programming (OOP)** é um paradigma de análise, projeto e programação de sistemas de software baseado na **composição e interação** entre diversas **unidades de software chamadas de objetos**.

# Programação Orientação a Objetos

- A análise e projeto orientados a objetos têm como meta identificar o **melhor conjunto de objetos** para descrever um sistema de software.
- O funcionamento deste sistema se dá através do **relacionamento e troca de mensagens** entre estes objetos.

# Programação Orientação a Objetos

- Na programação orientada a objetos, implementa-se um **conjunto de classes** que definem os objetos presentes no sistema de software.
- Cada classe determina o **comportamento** (definido nos métodos) e **estados possíveis** (atributos) de seus objetos, assim como o relacionamento com outros objetos.



**Classe:** representa um conjunto de objetos com características afins. Uma classe define o **comportamento dos objetos**, através de métodos, e quais estados ele é capaz de manter, através de atributos. Exemplo de classe: Os seres humanos.

**Subclasse:** é uma nova classe originada de sua classe pai.

**Objeto:** é uma **instância de uma classe**. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos.

Exemplo de objetos da **classe Humanos**: João, José, Maria.

**Atributos:** são **características de um objeto**. Basicamente a estrutura de dados que vai representar a classe.

O conjunto de valores dos atributos de um determinado objeto é chamado de estado.

Exemplos: **Funcionário:** nome, endereço, telefone, CPF, ....; **Carro:** nome, marca, ano, cor, ...; **Livro:** autor, editora, ano.

**Métodos:** definem as **habilidades** dos objetos.

- Bidu é uma **instância** da classe Cachorro, portanto tem habilidade para **latir**, implementada através do **método** `deUmLatido()`.
- Um **método em uma classe é apenas uma definição**.
- A ação só ocorre quando o **método é invocado através do objeto**, no caso Bidu.
- Dentro do programa, **a utilização de um método deve afetar apenas um objeto em particular**;
- Todos os cachorros podem latir, mas você quer que apenas Bidu dê o latido.
- Normalmente, uma **classe possui diversos métodos**, que no caso da classe Cachorro poderiam ser `sente()`, `coma()` e `morda()`.

**Herança (ou generalização)** é o mecanismo pelo qual uma classe (**sub-classe**) pode estender outra classe (**super-classe**), aproveitando seus comportamentos (**métodos**) e variáveis possíveis (**atributos**).

Há Herança múltipla quando uma **sub-classe possui mais de uma super-classe**. Essa relação é normalmente chamada de relação "é um". Um exemplo de herança: Mamífero é super-classe de Humano. Ou seja, um Humano **é um** mamífero.

**Abstração** é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais.

Ou seja, a complexidade foi “**escondida**” de você: isto é Abstração na prática!

**Polimorfismo** é a capacidade de um método poder ser implementado de diferentes formas, ou mesmo de realizar coisas diferentes.

Quando utilizamos Herança, teremos Classes filhas utilizando código comum da Classe acima, ou Classe pai.

Ou seja, as Classes vão compartilhar atributos e comportamentos (herdados da Classe acima).

Daí o nome: **Poli** (muitas) **morfismo** (formas).

# Exemplo

- **Classe** representa um conjunto de **objetos** com características afins. Uma classe define o comportamento dos objetos, através de métodos, e quais estados ele é capaz de manter, através de atributos.
- Exemplo de classe: Cachorro, Copo, Rádio, Telefone, etc.
- Tudo que representa uma categoria de objetos.

Classe	Objeto
Cachorro	Husky
	Cocker
	SRD
Rádio	AM-FM
	Bluetooth
	USB
Telefone	???
	???
	???



# Exemplo

- **Objeto** é uma instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos.

Classe	Objeto
Homem	João
	José
	Maria
Computador	Desktop
	Notebook
	???

# Exemplo

- **Atributos** são características de um objeto.
- Basicamente a **estrutura de dados** que vai representar a classe.

Objeto	Atributos
Funcionário	Nome, Endereço, Telefone, CPF.
Carro	Nome, Marca, Modelo, Ano, Cor.
Livro	Autor, Editora, Ano da Publicação

# Exemplo

- Os **atributos** possuem valores.

Objeto	Atributo	Valor
Carro	Nome	Gol
	Marca	VW
	Ano	2009
	???	???
Livro	Nome	Introdução à Orientação a Objetos com C++ e Python: Uma abordagem prática
	Autor	Orlando Saraiva Jr
	Editora	Novatec Editora
	???	???
Funcionário	Nome	Faustão
	Cargo	Apresentador
	Salário	R\$ 5 milhões

# Exemplo

**Métodos** definem as habilidades dos objetos. Um método em uma classe é apenas uma definição. A ação só ocorre quando o método é invocado através do objeto.

Objeto	Método
Cachorro	Acordar, Comer, Beber, Latir, Dormir.
Pessoa	Acordar, Correr, Estudar, Trabalhar, Falar, Gritar, Emocionar.
Carro	Ligar, Andar, ???
Rádio	Ligar, Tocar, Sintonizar, ???
Telefone	Discar, Tocar, ???

# Praticando os conceitos de OO

Uma empresa deseja realizar o **cadastro de funcionários** em um sistema próprio. Você foi contratado para desenvolver.

**Atenção para as Regra de Negócio**

**Campos de Entrada:**

Nome, Sexo, Nascimento, Endereço, Telefone, Data Admissão e Cargo.

**Base Cargos:**

Diretoria: R\$ 2000,00

Técnico: R\$ 1000,00

**Processamentos:**

- Calcula a idade
- Calcula o salário (cargo \* tempo admissão (anos))

**Campos de Saída:**

Idade, Salário

**Campos de ação:**

Cadastra, limpa e exibe

# Praticando os conceitos de OO

Desenvolva este projeto utilizando os conceitos de Orientação a objetos.

Requisitos Mínimos:

- **Classe** com Atributos e métodos;
- **Instância** de Objetos (Objeto);
- Para objetos do formulário, utilize os botões: **Cadastra**, **Limpa** e **Exibe**.

## Exemplo

Classe	Objeto	Atributos	Métodos
Formulário	Funcionário	Nome, Endereço ...	Cadastra, Limpa e Exibe

## Já disponível no class!

mãos à obra

# Introdução a Programação Orientação a Objetos I

Código da turma no classroom: iqmtxhd

Aulas: Segunda - 20:20 e Quinta - 18:30

[sarahsoares.com.br/](https://sarahsoares.com.br/)