



Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO

Programação Orientada a Objetos

Desenvolvimento de Sistemas

Prof. Me. Reneilson Santos

Fevereiro/2024

Agenda

- Programação Orientada a Objetos
- Conceitos POO
 - ◆ Objetos
 - ◆ Classes
 - ◆ Atributos e Métodos



Programação Orientada a Objetos

Programação Orientada a Objetos

A Programação Orientada a Objetos (POO) é um paradigma de programação amplamente utilizado na indústria de desenvolvimento de software. Ele se concentra na criação de objetos, que são instâncias de classes, e permite que os desenvolvedores **organizem** e **estruturem** seus programas de forma mais **intuitiva e modular**.



Objetivos da Orientação a Objetos

- O objetivo principal é lidar com a complexidade de se construir **softwares modernos**
- Esse objetivo está relacionado à **produção de softwares de qualidade**
- Implementações de software devem buscar **robustez, adaptabilidade e reusabilidade**

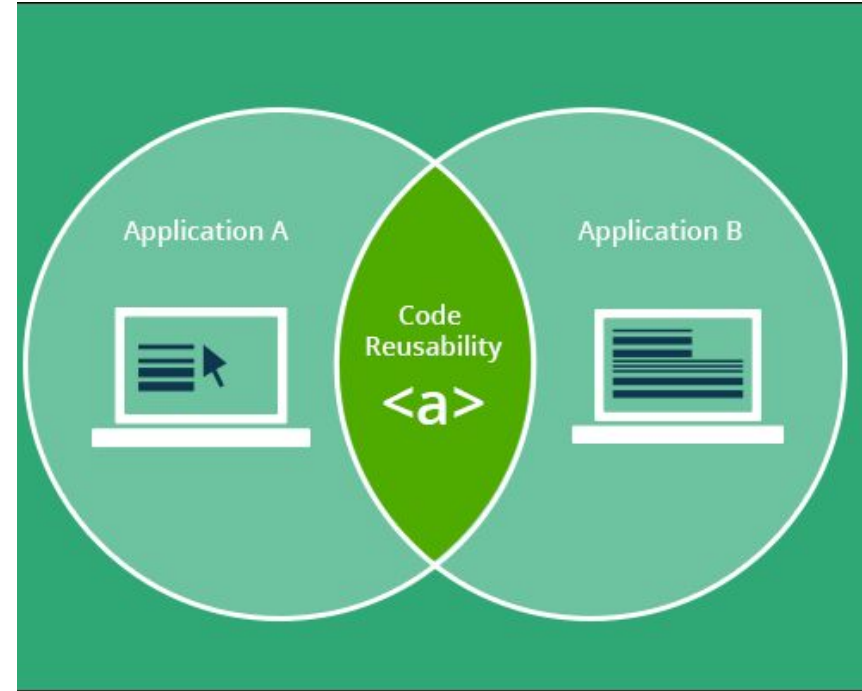
Por que POO?

A Programação Orientada a Objetos (POO) desempenha um papel fundamental no desenvolvimento de softwares modernos e é amplamente utilizada na indústria de tecnologia.

Sua importância deriva de diversos benefícios que ela oferece, os quais contribuem para tornar o desenvolvimento de software **mais eficiente, modular, sustentável e fácil de manter.**

Reutilização de Código

A POO permite criar classes que podem ser reutilizadas em diferentes partes do programa ou mesmo em projetos diferentes. Isso economiza tempo e esforço, pois desenvolvedores podem aproveitar classes já existentes em vez de reescrever o código do zero.



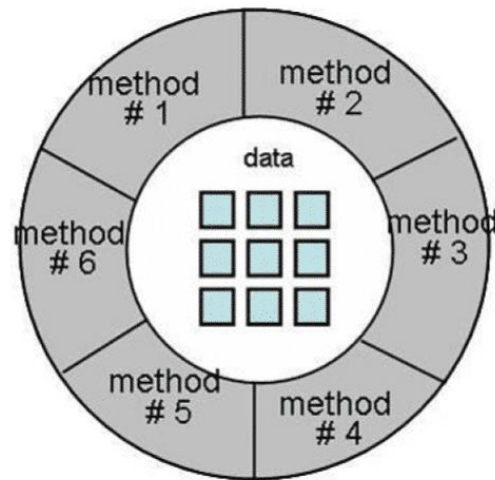
Modularidade

A abordagem orientada a objetos incentiva a divisão do programa em módulos independentes, representados por classes. Cada classe é responsável por uma tarefa específica, o que facilita a manutenção e o desenvolvimento incremental do software.



Encapsulamento e Ocultação de Dados

O encapsulamento permite ocultar detalhes de implementação dentro das classes. Isso protege os dados e evita que outras partes do programa acessem ou modifiquem informações de forma inapropriada, contribuindo para um código mais seguro.



Facilidade de Manutenção

Devido à modularidade e ao encapsulamento, alterações em uma parte do programa não devem afetar outras partes, desde que a interface pública das classes seja mantida inalterada. Isso facilita a manutenção do software e evita problemas em cascata.



Colaboração entre Equipes

A POO promove a divisão do trabalho em tarefas menores, facilitando o trabalho em equipe. Diferentes desenvolvedores podem trabalhar em classes distintas sem interferir no trabalho dos outros, desde que a interface pública seja respeitada.



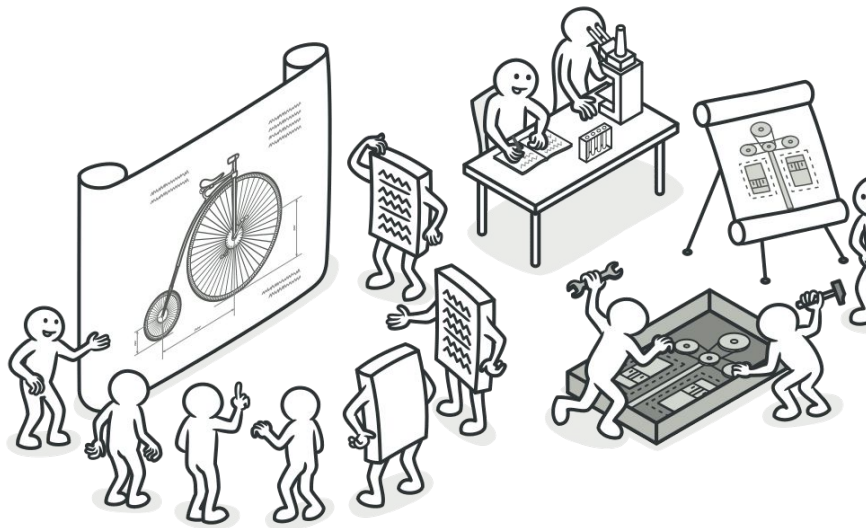
Escalabilidade

A estrutura orientada a objetos é altamente escalável, permitindo que o software cresça sem perder coesão ou aumentar o acoplamento. Isso é essencial para projetos que exigem expansão contínua ao longo do tempo.



Padrões de Design

A POO está intimamente relacionada a padrões de design de software, como o padrão de projeto (design pattern). Esses padrões são soluções comprovadas para problemas comuns e ajudam a desenvolver sistemas robustos e bem organizados.



Por que POO?

Em suma, a Programação Orientada a Objetos oferece uma abordagem **flexível, estruturada e orientada a problemas** para o desenvolvimento de software.

Ela permite criar sistemas mais **eficientes, fáceis de manter e capazes de acompanhar as necessidades em constante mudança** da indústria de tecnologia.

Com o crescimento contínuo da complexidade dos projetos, **a POO tornou-se um pilar essencial para a construção de softwares modernos e sustentáveis.**

Conceitos POO

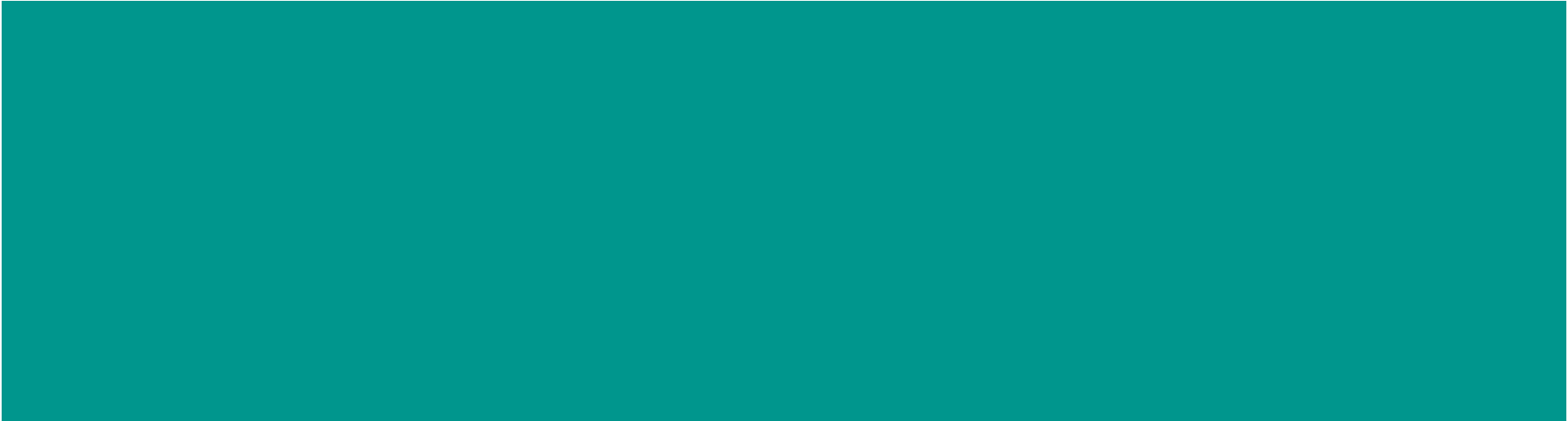
Conceitos de POO

Antes de iniciarmos a codificação em POO é importante conhecer alguns conceitos essenciais para construção de um código com o paradigma orientado a objetos.

Serão apresentados a seguir os conceitos de:

1. Objetos
2. Classes
3. Métodos

Objetos



Objetos

A POO se baseia na ideia de que um programa de computador pode ser modelado de forma **semelhante à realidade, representando entidades do mundo real** ou **abstrações** por meio de estruturas de dados e comportamentos associados.

Isso permite que os desenvolvedores organizem e interajam com os dados de uma maneira mais intuitiva e compreensível.

É nesse ponto que entra o objeto, que nada mais é do que essa representação da "coisa" real dentro do nosso sistema.

Representações do Mundo Real

Um objeto pode ser visto como uma representação de uma entidade ou objeto do mundo real.

Por exemplo, em um sistema de gestão de biblioteca, você poderia ter vários objetos "Livro", que representam a entidade "livro" no mundo real.

Características e Comportamentos

Os objetos têm características associadas a eles, também conhecidas como **atributos**.

Esses **atributos** representam **os dados que descrevem o objeto**.

Por exemplo, um objeto "Pessoa" pode ter atributos como nome, idade e endereço.

Além disso, os objetos têm comportamentos, que são definidos por meio de **métodos**.

Os **métodos** representam as **ações que os objetos podem realizar**.

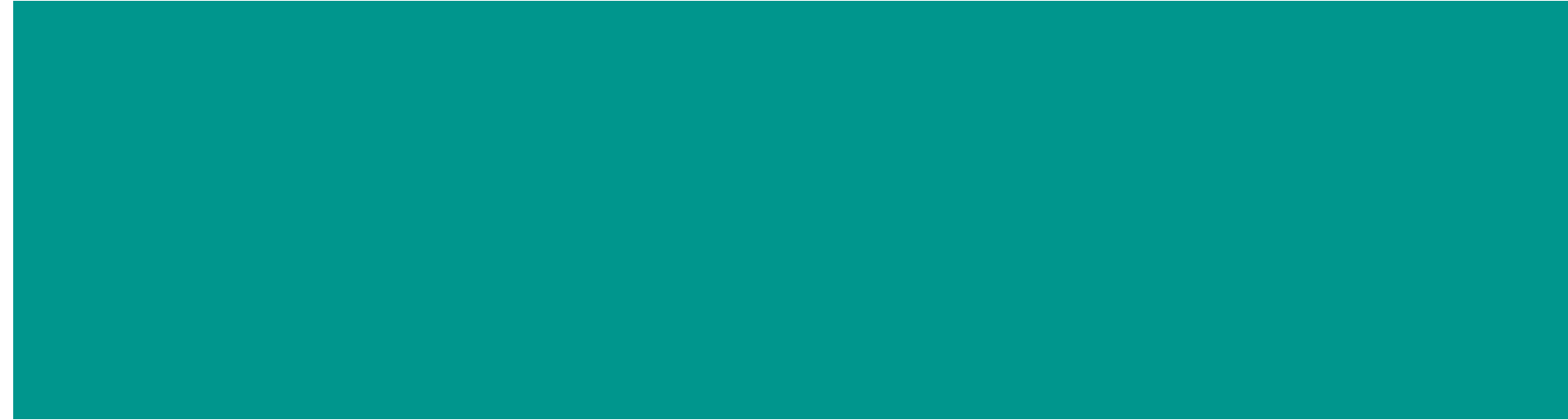
Por exemplo, um objeto "Carro" pode ter métodos como "ligar", "acelerar" e "frear".

Objetos

Em resumo, os objetos são elementos centrais na POO e permitem uma modelagem mais intuitiva e organizada de sistemas de software. Eles representam entidades do mundo real ou abstrações conceituais, possuindo atributos que descrevem suas características e métodos que definem suas ações.

Através da criação e manipulação de objetos, os desenvolvedores podem construir programas que são mais próximos da maneira como concebemos e interagimos com o mundo real.

Classes



Classe

Em Programação Orientada a Objetos (POO), uma classe é uma estrutura que descreve um tipo de objeto, definindo seus atributos (propriedades) e comportamentos (métodos).

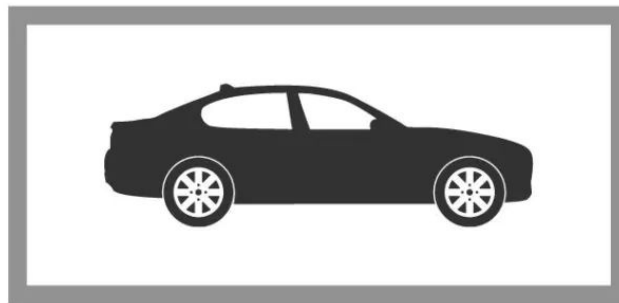
Ela serve como um modelo ou molde a partir do qual os objetos são criados.

Podemos pensar em uma classe como uma *planta* que contém todas as informações necessárias para construir objetos que se encaixem nessa descrição.

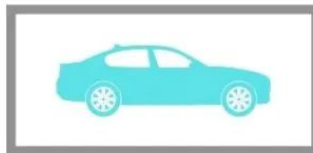
Classe



CLASSE CARRO

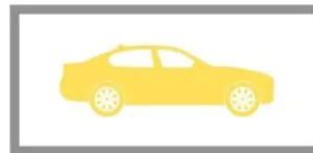


OBJETO



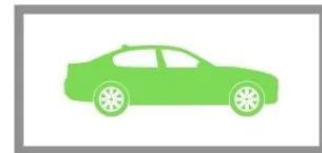
CELTA

OBJETO



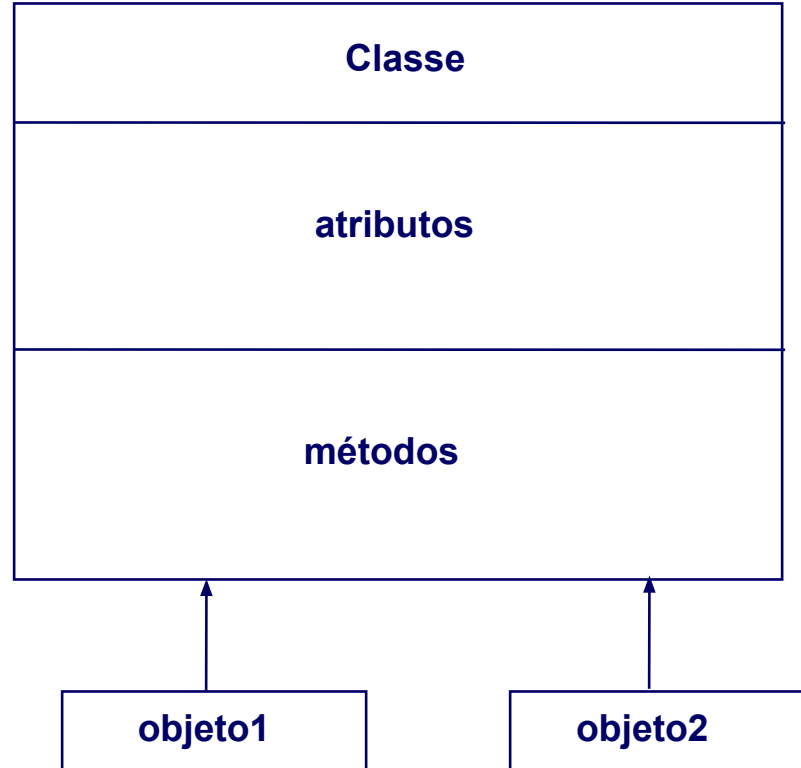
CORSA

OBJETO



FIAT UNO

Classes



Classes

Em resumo, as classes são blocos de construção essenciais da POO, permitindo que os programadores criem objetos com atributos e comportamentos específicos.

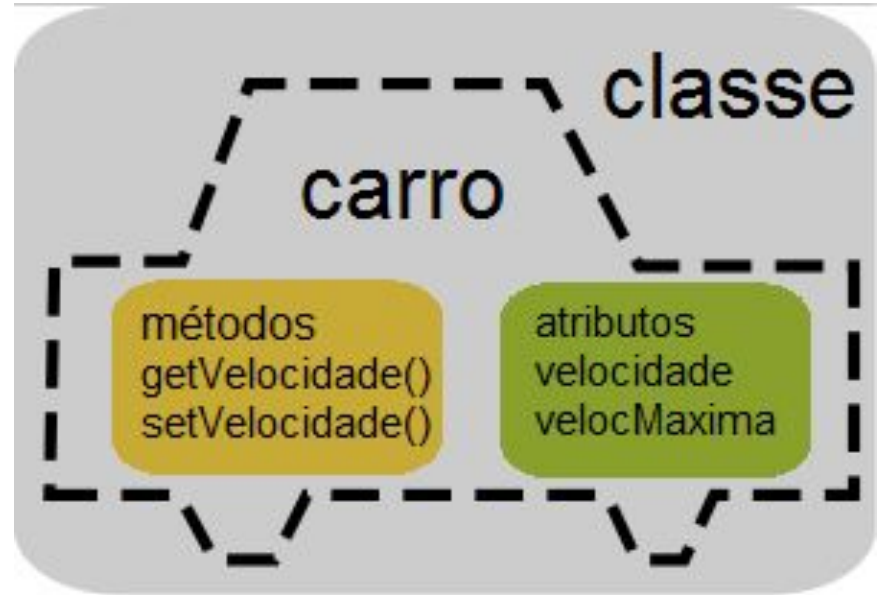
Elas são usadas para representar entidades do mundo real e abstrair conceitos em unidades coesas e reutilizáveis, tornando o código mais organizado, fácil de entender e modular.

Atributos e Métodos



Atributos e Métodos

Em Programação Orientada a Objetos (POO), atributos e métodos são componentes essenciais de um objeto e definem suas características (dados) e comportamentos (ações), respectivamente. Eles são definidos na classe, que é o modelo que descreve o objeto.



Atributos

Os atributos são as características ou propriedades do objeto.

Eles representam os dados que descrevem o objeto e o distinguem dos demais.

Na prática, os atributos são representados por variáveis que pertencem a uma classe.

Cada objeto criado a partir dessa classe terá seus próprios valores para essas variáveis.

Os atributos são usados para armazenar dados específicos de um objeto e para representar seu estado. Eles podem ser lidos e modificados.

Atributos

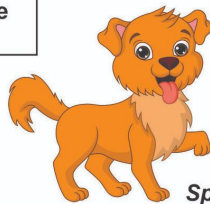
Classe Cães

Objetos Cachorros



Scooby

Cor
Peso
Altura
Nome



Spok

Cor
Peso
Altura
Nome

Cor
Peso
Altura
Nome



Malhado

Métodos

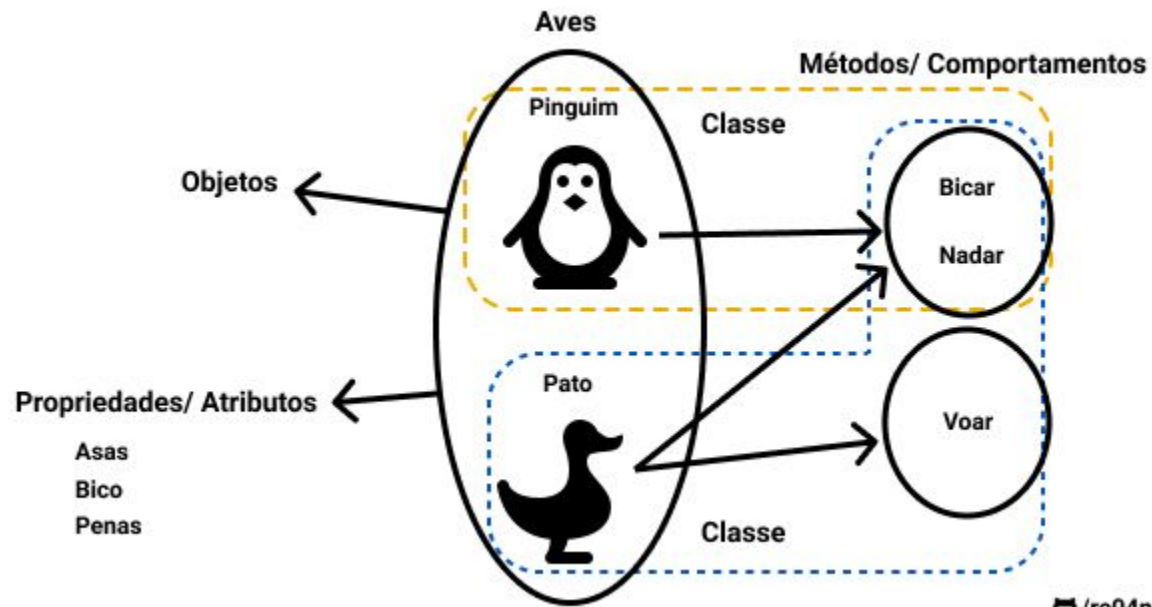
Os métodos são as ações ou comportamentos que um objeto pode realizar.

Eles representam as operações que podem ser executadas nos dados do objeto ou as interações que ele pode ter com outros objetos.

Os métodos são definidos dentro da classe e são acessados através dos objetos criados a partir dessa classe.

Os métodos são usados para realizar ações relacionadas a um objeto. Eles podem acessar e modificar os atributos do objeto e podem interagir com outros objetos, se necessário.

Métodos



Kahoot!

