

Programação Orientada a Objetos



Introdução

O que é programação orientada a objetos:

A programação orientada a objetos (POO) é um paradigma de programação que se baseia na organização do código em torno de objetos, que são instâncias de classes. Ela enfatiza o conceito de encapsulamento, onde os dados e comportamentos relacionados são agrupados em objetos, garantindo a integridade dos dados e facilitando a manutenção do código.

Principais características POO:

Classe

- Representação de um conjunto de objetos com características afins. Definição do comportamento dos objetos (métodos) e seus atributos (atributos).

Objeto

- Uma instância de uma classe.
- Armazenamento de estados através de seus atributos e reação a mensagens enviadas por outros objetos.

Herança

- Mecanismo pela qual uma classe (sub-classe) pode estender outra classe (super-classe), estendendo seus comportamentos e atributos.

Polimorfismo

- Princípio pelo qual as instâncias de duas classes ou mais classes derivadas de uma mesma super-classe podem invocar métodos com a mesma assinatura, mas com comportamentos distintos.

Encapsulamento

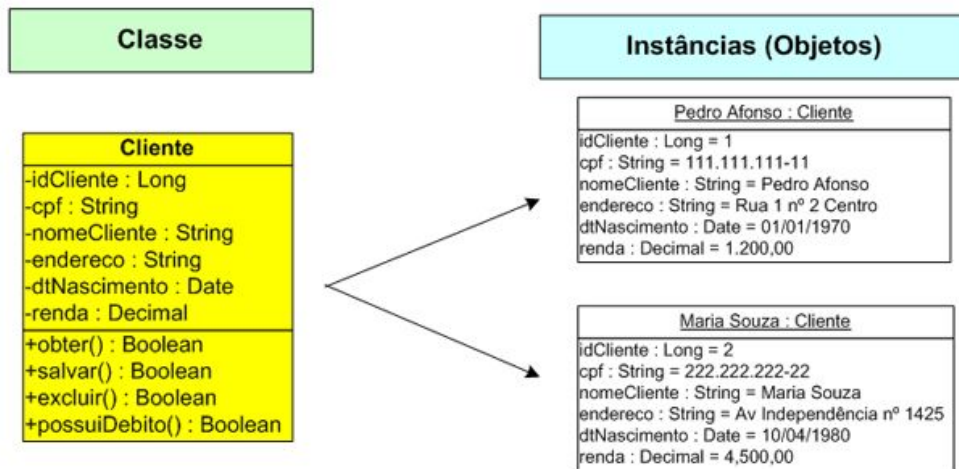
- Proibição do acesso direto ao estado de um objeto, disponibilizando apenas métodos que alterem esses estados na interface pública.

Classe:

Uma classe é uma estrutura fundamental na programação orientada a objetos (POO) que define um tipo de objeto. Ela encapsula atributos e métodos relacionados, fornecendo uma representação abstrata de um conceito no mundo real. Os atributos representam as características ou informações associadas a um objeto daquela classe, enquanto os métodos definem o comportamento ou as ações que o objeto pode realizar. Uma classe serve como um molde para criar múltiplas instâncias de objetos, promovendo a modularidade, a reutilização de código e o princípio do encapsulamento.

Objeto:

É basicamente uma instância de uma classe, funcionando como uma espécie de variável de um “tipo” que é a própria classe.



Objetos (Instâncias) da classe Cliente

Princípios POO:



Encapsulamento:

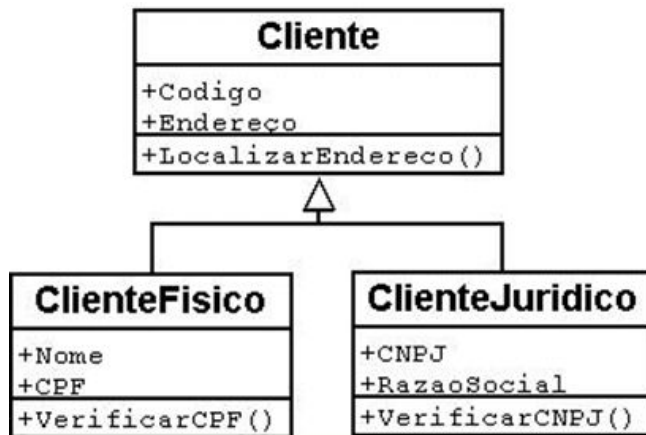
O encapsulamento é um princípio da programação orientada a objetos que envolve ocultar os detalhes internos de implementação de um objeto. Ele protege os dados e comportamentos de uma classe, permitindo acesso controlado apenas por meio de métodos públicos. O encapsulamento promove a segurança, a integridade dos dados e facilita a manutenção do código.

Abstração:

A abstração é um conceito-chave na programação orientada a objetos que permite simplificar a complexidade de um sistema ao focar apenas nos aspectos relevantes. Ela envolve a criação de classes abstratas, interfaces e métodos abstratos que definem apenas a estrutura e o comportamento essencial dos objetos. A abstração permite uma visão mais geral e organizada do código, facilitando a reutilização, manutenção e ampliação do software.

Herança:

A herança é um mecanismo na programação orientada a objetos que permite a criação de novas classes baseadas em classes existentes. Ela promove a reutilização de código ao herdar atributos e métodos da classe pai (superclasse). A herança permite criar hierarquias de classes, compartilhar comportamentos comuns e especializar as classes filhas (subclasses) com novos atributos e métodos.



Polimorfismo:

O polimorfismo é um princípio da programação orientada a objetos que permite tratar objetos de diferentes classes de forma uniforme. Ele permite que um mesmo método seja implementado de maneira diferente em diferentes classes, através da sobrescrita de métodos.

