Pilares do POO

Programação orientada a objetos (**POO**, ou **OOP** segundo as suas siglas em inglês), é um paradigma de programação baseado no conceito de "objetos", que podem conter dados na forma de campos, também conhecidos como *atributos* e códigos, na forma de procedimentos, também conhecidos como métodos.

Como se trata de, um contexto análogo ao mundo real, tudo no qual nos referimos são objetos, exemplo: Conta bancária, Aluno, Veículo, Transferência etc.

A programação orientada a objetos, é bem requisitada no contexto das aplicações mais atuais no mercado, devido a possibilidade de reutilização de código e a capacidade de representação do sistema, ser muito mais próximo do mundo real.

Para uma linguagem ser considerada orientada a objetos, esta deve seguir o que denominamos como ***Os quatro pilares da orientação a objetos***:

- **Encapsulamento:** Nem tudo precisa estar visível, grande parte do nosso algoritmo pode ser distribuído em métodos, com finalidades específicas que complementam uma ação em nossa aplicação.

  Exemplo: Ligar um veículo, exige muitas etapas para a engenharia, mas o condutor só visualiza a ignição, dar a partida e a *"magia"* acontece.
- **Herança:** Características e comportamentos comuns, podem ser elevados e compartilhados através de uma hierarquia de objetos.

Exemplo: Um Carro e uma Motocicleta possuem propriedades como placa, chassi, ano de fabricação e métodos como acelerar e frear. Logo, para não ser um processo de codificação redundante, podemos desfrutar da herança criando uma classe ***Veículo*** para que seja herdada por ***Carro*** e ***Motocicleta***.

- **Abstração:** É a indisponibilidade, para determinar a lógica de um ou vários comportamentos, em um objeto.

  Exemplo: ***Veículo***** ** determina duas ações como acelerar e frear, logo, estes comportamentos deverão ser *abstratos,* pois existem mais de uma maneira de se realizar a mesma operação. ver *Polimorfismo*.

- **Polimorfismo:** São as inúmeras maneiras de se realizar uma mesma ação.

  Exemplo: Veículo determina duas ações como acelerar e frear, primeiramente, precisamos identificar se estaremos nos referindo a ***Carro***** ** ou ***Motocicleta,*** para determinar a lógica de aceleração e frenagem dos respectivos veículos.

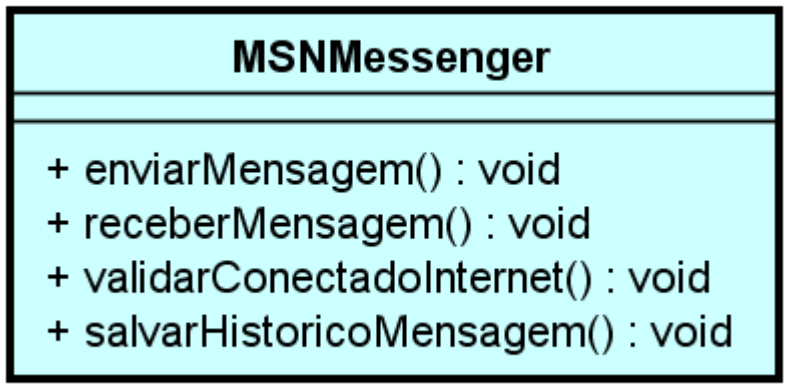
Em prática

Para ilustrar a proposta dos Princípios de POO, no nosso cotidiano, vamos simular algumas funcionalidades dos aplicativos de mensagens instantâneas pela internet.

MSN Messenger foi um programa de mensagens instantâneas criado pela Microsoft Corporation. O serviço nasceu a 22 de julho de 1999, anunciando-se como um serviço que, permitia falar com uma pessoa através de conversas instantâneas pela internet. Ao longo dos anos, surgiram novos serviços de mensagens pela internet, como **Facebook Messenger** e o **VKontakte Telegram**.

Vamos descrever em UML e depois em código, algumas das principais funcionalidades de qualquer serviço de comunicação instantânea pela internet, inicialmente pelo MSN Messenger e depois inserindo os demais, considerando os princípios de POO.

UML | Code



Pontos de atenção:

- Todos os métodos da classe são **public** (tudo realmente precisa estar visível ?);
- Só existe uma única forma de se comunicar via internet (como ter novas formas de se comunicar mantendo a proposta central ?).

Previous
UML

Next
Encapsulamento

Last updated 1 year ago