

Programação Orientada a Objetos (POO)

A Programação Orientada a Objetos (POO) surgiu com a finalidade de facilitar a vida daqueles que trabalham com desenvolvimento de software, pois na POO o difícil não é desenvolver bem um software, mas sim desenvolver um software que satisfaça o cliente, ou seja, garantir que o que será entregue será realmente o que foi pedido.

Uma das características da POO é fazer com que o programador pense as coisas de forma distintas, transformando-as assim em objeto, aplicando propriedades e métodos, que comentaremos mais adiante, reduzindo assim a complexidade no desenvolvimento e manutenção de software, aumentando a produtividade.



Características - POO

A POO trabalha utilizando classes, ela é quem dá a vida ao objeto, pois nela é que são criadas as funcionalidades através dos métodos, dizendo se o objeto vai andar parar rodar, etc., e suas especificidades através dos atributos, sua cor, seu nome, etc.. a classe não é o objeto, ela é utilizada para construí-lo.

O objeto é uma abstração de software que pode ser algo representado por algo real ou virtual, que é formado por um conjunto de propriedades (variáveis) e procedimentos (métodos), onde as variáveis possuem um tipo que define os possíveis valores que pode representar um número inteiro, real ou string e, os métodos são rotinas que, quando executadas realizam algumas tarefas como alterar o conteúdo de uma variável do objeto.

A classe é a abstração do objeto. Na realidade ao programarmos um objeto, definindo suas características e funcionalidades, estamos programando uma classe.

Dentre alguns recursos aplicados na programação orientada a objeto, temos: encapsulamento, herança e polimorfismo.



Encapsulamento - POO

Encapsulamento é a limitação imposta a atributos de uma classe, restringindo o acesso aos métodos dessa classe através de limitadores de acesso, dos quais os mais utilizados são:

Público (public) – a classe só é vista pelas demais classes se estiverem dentro do mesmo pacote;

Protegida (protected) – o acesso se restringe apenas a própria classe e suas subclasses;

Privado (private) – o acesso só é permitido somente pela própria classe.



Herança - POO

O conceito de herança nos permite definir uma nova classe, com base em uma já existente. A classe criada (subclasse ou classe derivada) automaticamente herda todas as variáveis e métodos da classe já existente (superclasse). O mecanismo de herança permite que a subclasse inclua ou sobreponha novas variáveis e métodos da superclasse. O mecanismo de herança é recursivo, permitindo criar-se uma hierarquia de classes, do nível mais alto, onde as características herdadas são comuns a todos os objetos desta classe até os níveis inferiores, onde estão as especializações das classes superiores. As subclasses herdam as características comuns, definindo suas propriedades específicas, ou seja, facilita a construção de novos objetos a partir de outros já existentes, sem a necessidade de reescrevê-lo todo.

Polimorfismo - POO

Polimorfismo: A palavra polimorfismo utilizada na biologia para definir várias formas de membros de uma mesma espécie.

Utilizando essa analogia, o mecanismo de polimorfismo utilizado na POO, nos permite tratar objetos semelhantes de uma maneira uniforme.

Para que o polimorfismo seja utilizado, é necessária a aplicação do conceito de herança, aplicados somente aos métodos da classe.

Anterior a utilização da POO, sempre que uma nova funcionalidade fosse criada ou acrescentada, a aplicação deveria ser alterada e recompilada, porém com a utilização do polimorfismo, acrescentar novos métodos a classe já existente sem a necessidade de ter que recompilar a aplicação tornou-se mais fácil.



Polimorfismo - POO

Com base nos conceitos de objetos, classes, encapsulamento, herança e polimorfismo, que são considerados como o paradigma da POO representando uma forma inovadora de pensar e criar novos softwares, trazendo inúmeros benefícios e valores agregados no desenvolvimento deles, tais como: a reutilização de códigos que reduz o tempo e o risco no desenvolvimento, aumentando assim a confiabilidade do software, haja vista que, os códigos reutilizados já foram testados, facilita também a compreensão da linguagem com desenvolvedor na hora da manutenção de softwares complexos.



Conclusão

A linguagem de POO está em nosso dia-a-dia facilitando as nossas vidas. Muitas das vezes estamos desfrutando dos seus benefícios sem mesmo perceber, quando por exemplo: quando falamos ao celular, utilizamos o PC e, até mesmo quando estamos fazendo uma simples compra no supermercado, não temos essa percepção mas, a maioria dos programas instalados nesses equipamentos foi desenvolvido em linguagem de POO.

A linguagem de POO possui ainda inúmeras ferramentas e aplicativos facilitadores que interagem ajudando os desenvolvedores e, apresenta aos iniciantes um leque de opções, dando a oportunidade de se identificarem com a linguagem de POO ou de continuar a procura de uma nova ferramenta de estudo ou de trabalho. A linguagem de POO não significa a reinvenção da roda mas, a mão na roda para qualquer desenvolvedor.

