Conhecendo a programação orientada a objeto

Orientação a objeto é um modelo de programação pensado a mais tempo do que se imagina. Teve real implementação na linguagem Simula 67 e se popularizou com a com a linguagem Smalltak, criado por Alan Kay, mesmo criador do termo programação orientada a objetos. Somente a poucos anos esse modelo foi aceito pelas grandes corporações e linguagem como Java, C++, C# e Python adotaram essa ideia. Diferente das metodologias procedurais que utilizam o conceito de um processador, uma memória e dispositivos de I/O para processar, armazenar e exibir as informações, a OOP emprega um conceito mais real, mais concreto, que é o de Objeto, utilizado para simular o melhor possível do que seria o mundo real no virtual.

Esse modelo gira em torno do que diz no próprio nome, o objeto. Pode ser definido como um elemento ativo e com características que o tornam inteligente, podendo tomar decisões quando necessário.

O programador de OO tem por obrigação moldar todos esses objetos, que fazem a similaridade com o mundo real. Eles devem interagir entre si e fazer isso por meio de troca de mensagens e o programador quem especifica quem deve receber, quais mensagens enviar e o que deve acontecer quando recebida uma determinada mensagem por um determinado objeto.

A mensagem é um texto sem espaços que os objetos entendem e junto com ela pode-se passar parâmetros, que são informações sobre o objeto. Assim os objetos conseguem trocar informações entre si facilmente.

Para exemplificar e mostrar melhor, vamos a um exemplo prático: imagine que você está desenvolvendo um software para uma locadora e esta locadora tem diversos clientes. Como estamos tentando modelar um sistema baseado no sistema real, nada mais obvio do que existirem objetos do tipo Clientes dentro do nosso programa, e esses Clientes dentro do nosso programa nada mais serão do que objetos que "simulam" as características e ações no mundo virtual que um cliente pode realizar no mundo real.

Na programação orientada a objetos existe também o conceito de classes, que são a abstração que define um tipo de objeto e o que esses objetos possuem dentro deles, seus atributos, como também definem as ações que esses objetos podem realizar, seus métodos. As classes somente são um conceito, diferente dos objetos que representam coisas do mundo real. Por exemplo a Classe Cliente especifica o nome de cada cliente e que ele pode comprar certos produtos da loja, mas não pode realizar a ação de comprar, já os objetos do tipo Cliente podem realizar a compra. Nesse caso o nome do cliente seria o atributo da classe e a ação seria um método, dito anteriormente.

Outro conceito são os construtores, que servem para inicializar os atributos de uma classe e é executado automaticamente quando você cria um novo objeto.

Existem alguns outros conceitos básicos e fundamentais para a formação de uma linguagem OO que são abstração, encapsulamento, herança e polimorfismo.

**Abstração** é considerada como a habilidade de modelar características do mundo real do problema que o programador esteja tentando resolver.

**Encapsulamento** contribui fundamentalmente para diminuir os malefícios causados pela interferência externa sobre os dados. Partindo desse princípio, toda e qualquer transação feita com esses dados só pode ser feita através de procedimentos colocados "dentro" desse objeto, pelo envio de mensagens. Desta maneira, dizemos que um dado está encapsulado quando envolvido por código de forma que só é visível na rotina onde foi criado.

O conceito de **Herança** basicamente diz que uma classe pode se estender a outras, podendo herdar suas características e possuir, ainda, suas próprias. Como por exemplo a Classe Cliente pode herdar o atributo sexo da Classe Pessoa, mas a Classe Cliente tem o atributo matrícula que a Classe Pessoa não possui.

**Polimorfismo** em orientação a objetos, significa que um mesmo tipo de objeto, sob certas condições, pode realizar ações diferentes ao receber uma mesma mensagem. Ou seja, apenas olhando o código fonte não sabemos exatamente qual será a ação tomada pelo sistema, sendo que o próprio sistema é quem decide qual método será executado, dependendo do contexto durante a execução do programa.

Todos esses conceitos serão usados por uma linguagem OO e é um essencial conhecimento para começar a programar em uma delas, pois, entendendo-se esses conceitos, você encontrará bem menos obstáculos na hora de resolver um problema. Isso é só uma breve explicação e a muito mais a especificar se quiser se aprofundar nesse ótimo modelo de programação.