



Orientação a Objetos

É o nome que os pesquisadores da computação, na década de 70, usaram para batizar esse novo paradigma¹. Esse nome foi dado porque, de acordo com a visão de mundo desses pesquisadores, tudo é um objeto, ou ao mínimo, se pretende que seja.

A proposta da Orientação a Objetos é permitir que os programadores organizem os programas da mesma forma que as nossas mentes enxergam os problemas e/ou as diversas situações a nossa volta: não como um conjunto de espaços de memória ou estruturas complexas de dados, mas como um conjunto de “coisas” que fazem parte do problema. “Coisas” que fazem parte de um determinado contexto e que se relacionam entre si de diversas maneiras. Essas “coisas”, dentro do paradigma da Orientação a Objetos, são conhecidas e entendidas como objetos.

Orientação a objeto é um conceito que está relacionado com a ideia de **classificar, organizar e abstrair coisas**. Veja a definição formal:

"O termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados."

Falando uma linguagem mais simples para isto, vamos para um ambiente que se conhece bem: **A casa !**

Agora, olhando para a estante, o guarda-roupa, o armário, a cozinha, em todos estes lugares existe uma classificação das coisas e , somente de olhar para eles, já se sabe relacionar a classificação utilizada em cada um destes nestes lugares.

Na estante agrupa-se e organiza-se os livros, no guarda roupa as camisas, calças , meias, ternos, etc. Todos os objetos classificados nestes lugares foram organizados baseado em alguma concepção que se possuía sobre eles.

No contexto orientado a objeto **a estante , o armário, a cozinha** são chamados de **classes**.

No contexto de software podemos dizer que :

¹ Um paradigma é uma maneira diferente de se tentar enxergar uma mesma coisa, um mesmo mundo.



“Uma classe é um gabarito para a definição de objetos. Através da definição de uma classe, descreve-se que propriedades (ou atributos) o objeto terá.”

Uma classe mantém dois elementos importantes : **estrutura e comportamento**.

- Uma estrutura representa os **atributos** que descrevem a classe.
- Um comportamento representa os **serviços** que a classe suporta.

Na '**classe**' do guarda-roupa, uma camisa amarela pode ser colocada em uma outra classe. A '**classe**' camisa.

- Cada camisa tem uma estrutura que é : *a textura, a cor, o tamanho e o modelo*.
- Cada camisa tem um comportamento que é : *ordenar, rasgar, desbotar*.

Ao entender que existe uma **classe** camisa, entende-se que existem diversos tipos de camisas com suas características, ou seja, existem diversos objetos camisas que podem ser criados a partir da classe camisa. Daí tem-se o conceito de **objetos**.

O conceito de orientação a objetos é abordado desta mesma maneira sempre. No contexto do software podemos então dizer que :

*“É através de **objetos** que (praticamente) todo o processamento ocorre em aplicações desenvolvidas com linguagens de programação orientadas a objetos.”*

- Primeiro classifica e abstrai os elementos no sistema para proporcionar uma certa ordem, e, ao fazer isto, define-se uma classe ;
- Feita a definição da classe, pode-se criar objetos desta classe. (**instanciar**)

Exemplo clássico da planta de uma casa.

- A **Classe** seria um gabarito (**como uma planta de uma casa**).
- O **objeto** é concretização do gabarito (**casas feitas a partir da mesma planta**).

Quando leva-se estes conceitos para as linguagens de programação nada se altera.

Existem alguns conceitos básicos que estão vinculados ao conceito de orientação a objetos. São eles:

- Abstração;
- Hierarquização;
- Encapsulamento;
- Polimorfismo.



Abstração

Abstração é a capacidade de ignorar aspectos não relevantes de um assunto para o propósito em questão, tornando possível uma concentração maior nos assuntos principais. Quando se quer diminuir a complexidade, seleciona-se parte do que está sendo analisado, em vez de tentar compreender o todo.

Selecionam-se os objetos que fazem parte do sistema que está sendo analisado. São eles que vão definir o escopo do sistema. Existe uma grande dificuldade em saber quais objetos fazem parte do sistema em questão e quais não fazem parte. Para que isso seja bem feito, é necessário ter um bom conhecimento do negócio, fixar bem os objetivos do sistema e a quem esse atinge. Selecionados os objetos, é necessário definir os procedimentos desses objetos, ou seja, definir o comportamento de cada um. Isto vem de acordo com o escopo do sistema. Só serão definidos os procedimentos que ajudarão a resolver o problema pelo qual o sistema está sendo criado. Definir procedimentos errados pode significar que o escopo do sistema e seus objetivos não foram bem entendidos.

Hierarquização

A herança ou hierarquização é um mecanismo que permite que uma classe possa herdar o comportamento de outra classe, ao mesmo tempo em que novos comportamentos podem ser estabelecidos. A vantagem da herança é agrupar coisas comuns para poder reaproveitar código. As sub-classes (classes filhas) herdam da super-classe (classe pai). Ou seja, é a criação de uma nova classe a partir de uma classe existente. Exemplo: Pessoa é um tipo geral de objeto. Empregado e Estudante, um tipo mais específico de Pessoa. Assim, podemos dizer que Empregado e Estudante são subtipos de Pessoa, ou de forma reversa, Pessoa é um supertipo de Empregado ou Estudante. Todavia, hierarquizar um todo não significa priorizar determinada parte.

Encapsulamento

O encapsulamento é uma forma de esconder detalhes de funcionamento do programa e também uma maneira de restringir o acesso a certas propriedades ou componentes. É fundamental para permitir que o programa seja suscetível a mudanças. Encapsulamento não é esconder os dados, mas o encapsulamento leva à ocultação de dados.



ESCOLA ESTADUAL DE EDUCAÇÃO PROFISSIONAL DR. SOLON TAVARES

Técnico em Informática

Introdução a Orientação a Objetos- Profª Ingrid Santos

Polimorfismo

Assim como na biologia, o polimorfismo se refere à capacidade de assumir diferentes formas ou estágios. Em orientação a objetos, polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse.



ESCOLA ESTADUAL DE EDUCAÇÃO PROFISSIONAL DR. SOLON TAVARES

Técnico em Informática

Introdução a Orientação a Objetos- Profª Ingrid Santos

FONTES UTILIZADAS

Fundamentos de Orientação a Objetos e UML. Universidade do Vale dos Sinos (2007).

JUNIOR, Wilson. *Conceitos Fundamentais de Orientação a Objetos*. In: blog.wilsonjunior.net. Disponível em: <<https://blog.wilsonjunior.net/conceitos-fundamentais-de-orientacao-a-objetos/>>.

MACORATTI, José Carlos. *O que significa Orientação a objetos?*. In: Macoratti.net. Disponível em: <http://www.macoratti.net/oo_conc2.htm>.

RICARTE, Ivan Luiz Marques. *Polimorfismo*. In: Unicamp.br. Disponível em: <<http://www.dca.fee.unicamp.br/cursos/PooJava/polimorf/index.html#:~:text=Polimorfismo%20%C3%A9%20o%20princ%C3%ADpio%20pelo,objeto%20do%20tipo%20da%20superclasse.>>