

Programação Orientada a Objetos

PARADIGMAS DE PROGRAMAÇÃO

O que é Orientação a Objetos?

- Orientação de objetos (OO), em uma definição formal, é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos. Ou seja, é um modelo utilizado no desenvolvimento de software onde trabalhamos com unidades chamadas objetos.
- Como veremos, a orientação a objetos visa um pensamento o mais próximo possível da vida real. Ao programar orientado a objetos não teremos apenas funções que são chamadas uma seguida da outra, mas sim objetos que possuem estado e comportamento.

O que é um Objeto?

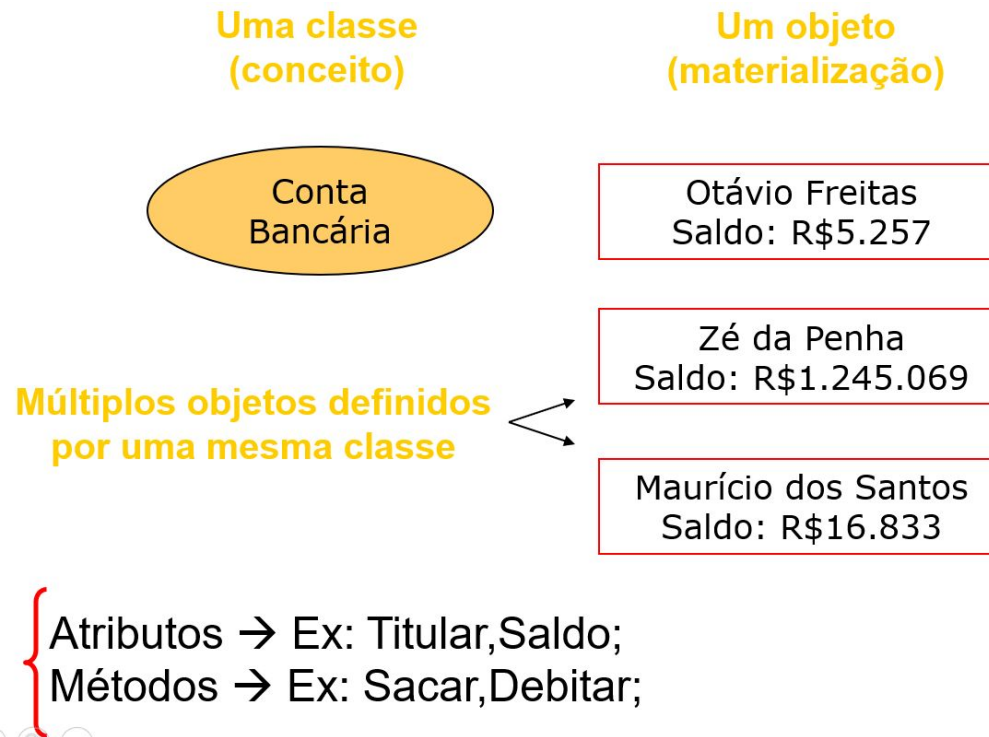
- Um objeto possui:
 - Estados: características;
 - Comportamentos: o que pode ser feito com ele (ou nele);
- Por exemplo:
 - Os estados de uma conta bancária são o seu número e o saldo;
 - Os comportamentos atribuídos a uma conta bancária são a habilidade de fazer depósitos e saques;
- Note que um comportamento de um objeto pode modificar seus estados;

Classe

- Um objeto é definido por uma classe.
- A classe usa métodos para definir o comportamento do objeto e atributos para definir as características do objeto;
- Múltiplos objetos podem ser criados por uma única classe;

{
Classe → conceito;
Objeto → materialização

Classes e Objetos



Criando Objetos

- Uma variável pode guardar um tipo primitivo ou uma referência para um objeto;
- Usa-se **new** para criar um novo objeto:

```
Arvore carvalho = new Arvore();
```

Isto chama o *construtor* de Arvore, que é um 'método' especial que constrói o objeto e possui o mesmo nome que sua classe;

- O ato de criar um objeto é chamado de instanciação;
- Um objeto é uma instância de uma classe particular;

Indo para o Java

- Vamos pensar em um sistema simples, um aparelho de ar-condicionado, quais são os atributos de um ar-condicionado? Quais são os métodos de um ar-condicionado?
- Agora vamos aumentar nosso modelo, queremos também uma televisão, que métodos e atributos tem uma televisão?
- Mas eles não tem algo em comum?
Passemos para os 4 pilares da POO.



Abstração

- A abstração consiste em um dos pontos mais importantes dentro de qualquer linguagem Orientada a Objetos. Como estamos lidando com uma representação de um objeto real (o que dá nome ao paradigma), temos que imaginar o que esse objeto irá realizar dentro de nosso sistema.

Encapsulamento

- O encapsulamento é uma das principais técnicas que define a programação orientada a objetos. Se trata de um dos elementos que adicionam segurança à aplicação em uma programação orientada a objetos pelo fato de esconder as propriedades, criando uma espécie de caixa preta.
- A maior parte das linguagens orientadas a objetos implementam o encapsulamento baseado em propriedades privadas, ligadas a métodos especiais chamados getters e setters, que irão retornar e setar o valor da propriedade, respectivamente. Essa atitude evita o acesso direto a propriedade do objeto, adicionando uma outra camada de segurança à aplicação.

Herança

- O reuso de código é uma das grandes vantagens da programação orientada a objetos. Muito disso se dá por uma questão que é conhecida como herança. Essa característica otimiza a produção da aplicação em tempo e linhas de código.
- Para entendermos essa característica, vamos imaginar uma família: a criança, por exemplo, está herdando características de seus pais. Os pais, por sua vez, herdam algo dos avós, o que faz com que a criança também o faça, e assim sucessivamente.

Polimorfismo

- Outro ponto essencial na programação orientada a objetos é o chamado polimorfismo. Na natureza, vemos animais que são capazes de alterar sua forma conforme a necessidade, e é dessa ideia que vem o polimorfismo na orientação a objetos. Como sabemos, os objetos filhos herdam as características e ações de seus “ancestrais”. Entretanto, em alguns casos, é necessário que as ações para um mesmo método seja diferente. Em outras palavras, o polimorfismo consiste na alteração do funcionamento interno de um método herdado de um objeto pai.

Pensando em Herança

