

Programação Orientada a Objetos

Conceitos Básicos

Cristiano Lehrer, M.Sc.

Introdução (1/3)

- O conceito de **Programação Orientada a Objetos** tem suas raízes na SIMULA 67, mas não foi amplamente desenvolvido até que a evolução da Smalltalk resultasse na produção de sua versão 80, em 1980.
- Uma linguagem orientada a objetos deve oferecer três recursos chaves:
 - Tipos de dados abstratos.
 - Herança.
 - Um tipo particular de vinculação dinâmica.

Introdução (2/3)

- A **programação orientada por procedimentos** (década de 70), concentra-se em subprogramas e em bibliotecas de subprogramas:
 - Se um *array* de valores inteiros precisa ser classificado, ele é enviado como um parâmetro a um subprograma que o classifica.
- A **programação orientada a dados** (década de 80), concentra-se em tipos de dados abstratos:
 - Se um objeto *array* de valores inteiros precisa ser classificado, chama-se o subprograma definido no objeto *array* específico.

Introdução (3/3)

- Programação orientada a dados é igual a programação orientada a objetos?
 - Na programação orientada a dados, os objetos:
 - Não são extensíveis, ou seja, não é possível fazer alterações incrementais em um objeto sem abrir o código a força e fazer as alterações diretamente.
 - Não podem ser baseados em outros objetos, a não ser através de delegação.
 - Não podem compartilhar o tipo de outro objeto.
- A **programação orientada a objetos** dá o próximo passo lógico após a programação orientada a dados, adicionando herança e polimorfismo.

Domínio

- O **domínio** é o espaço onde um problema reside, sendo o conjunto de conceitos que representam os aspectos importantes do problema que se está tentando resolver.
- Por exemplo, considere a pessoa Fulano:
 - No papel de aluno numa universidade, as informações mais significativas de Fulano são as referentes a sua condição acadêmica.
 - No papel de paciente num hospital, as informações mais significativas de Fulano são as referentes a sua condição clínica.
 - No papel de correntista num banco, as informações mais significativas de Fulano são as referentes a sua condição financeira.

Objeto

- Um **objeto** é uma construção de software que encapsula estado e comportamento.
- Os objetos permitem que o software seja modelado em termos reais e abstrações.
- Em uma linguagem de programação orientada a objetos pura, tudo é um objeto, desde os tipos mais básicos, como inteiros e lógicos, até as instâncias de classes mais complexas.
 - Smalltalk é uma linguagem orientada a objetos pura.
- Nem todas as linguagens de programação orientadas a objetos chegam a esse ponto, como o caso do Java e do C++, onde as primitivas não são tratadas como objetos.

Classe

- Uma **classe** define os atributos e comportamentos comuns compartilhados por um tipo de objeto.
- Os objetos de certo tipo ou classificação compartilham os mesmos comportamentos e atributos.
- As classes atuam de forma muito parecida com um cortador de molde ou biscoito, no sentido de que você usa uma classe para criar ou **instanciar** objetos.
- Por exemplo, uma classe Pessoa pode ter os objetos Fulano, Ciclano e Beltrano, todos compartilhando os mesmos comportamentos e atributos.

Atributos

- **Atributos** são as características de uma classe visíveis externamente.
- Por exemplo, a cor dos olhos e a cor dos cabelos são exemplos de atributos de uma classe Pessoa.
- Um objeto pode expor um atributo fornecendo um link direto a alguma variável interna, ou retornar o valor do atributo através de um método.
- Em termos de programação, atributos são as variáveis de uma classe/objeto.

Comportamento

- Comportamento é uma ação executada por um objeto quando passada uma mensagem ou em resposta a uma mudança de estado:
 - É algo que um objeto faz.
- Considerando novamente a classe Pessoa, podemos considerar que andar, correr e pular são comportamentos (ações) que uma pessoa pode fazer.
- Em termos de programação, o comportamento são os métodos/procedimentos que a classe/objeto possui.

Métodos

- **Construtores** são métodos usados para inicializar os objetos durante sua instanciação.
 - A criação de objetos é chamada de instanciação porque ela cria uma instância do objeto da classe.
- Os **acessores** (*get*) dão acesso aos dados internos de um objeto.
 - Entretanto, os acessores ocultam o fato de os dados estarem em uma variável, em uma combinação de variáveis ou serem calculados.
- Os **mutantes** (*set*) permitem a alteração do estado interno de um objeto.