VOLTAR

Conceitos de Orientação a Objetos

Esse tópico discute os conceitos básicos para a interpretação do que é possível se realizar com a Análise Orientada a Objetos.

NESTE TÓPICO

Marcar tópico

A análise orientada a objetos baseia-se em alguns princípios que são utilizados para administrar a complexidade e facilitar a modelagem de sistemas. O entendimento desses princípios é essencial para a interpretação e uso de sistemas modelados orientados a objeto.

São quatro os princípios da orientação a objetos: abstração, herança, encapsulamento e polimorfismo, sendo que esse tópico traz mais destaque para os três últimos. Porém, antes mesmo de discutir os princípios, se faz necessário definir algumas questões, conforme mostrado abaixo:

- Objeto: Qualquer coisa visível ou tangível para qual a ação, pensamento ou sentimento é direcionado, ou seja, o mundo real é formado por coisas e para a orientação a objetos estas coisas são denominadas de objetos.
- Classe: Representação de um conjunto de objetos que possuem os atributos (propriedades) e comportamentos (métodos) semelhantes.
- Instância: Objeto que foi criado em função de uma determinada classe.

Princípio da Herança

A palavra herança traz à lembrança de recepção de algo vindo de um ancestral. É exatamente esse o conceito aplicado quando utilizamos herança na orientação a objetos. O que a herança permite dentro da orientação a objetos é que uma classe Filha possa herdar Atributos e/ Comportamentos de uma classe Pai.

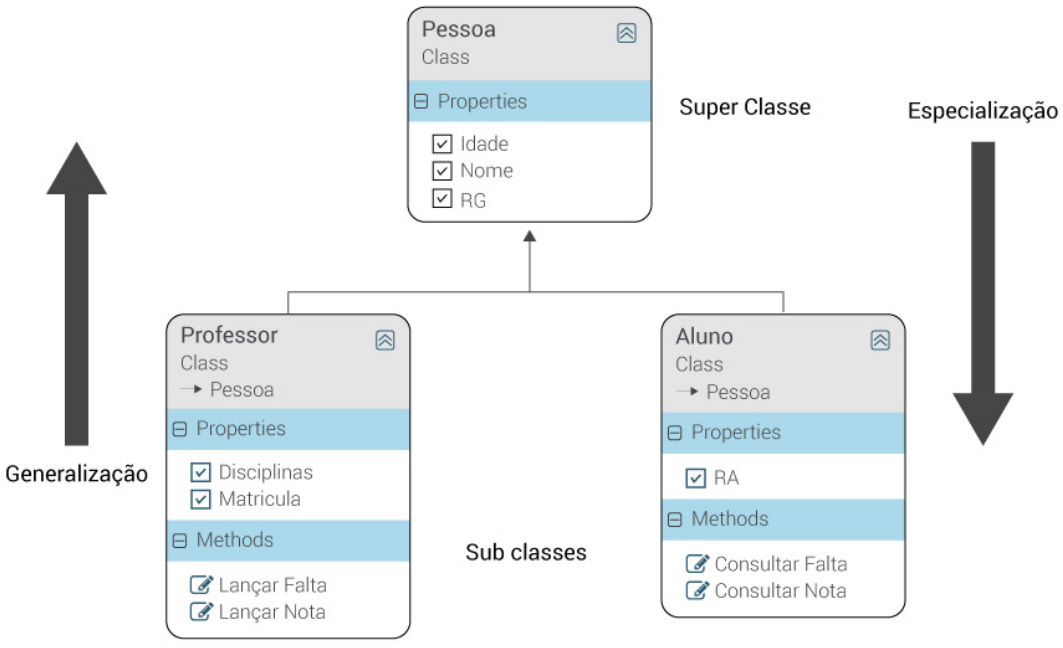
Existem dois tipos de herança: Simples e Múltipla. Um grande exemplo de herança múltipla somos nós, afinal adquirimos características e comportamentos de nossos pais e mães. Entretanto, por questões de complexidade, a herança mais utilizada nas linguagens de programação e, por consequência, na modelagem de sistemas orientados a objetos, é a Herança Simples, onde um pai pode ter inúmeros filhos, mas os filhos são provenientes de um único objeto Pai.

Também é possível observar diferentes maneiras de se descrever uma herança entre duas classes. Vamos tomar como exemplo a classe "A" e "B", sendo que "A" herda características e/ou comportamentos da classe "B". Pode-se dizer que:

- A classe Filha "A" herda da classe Pai "B", ou
- A classe Filha "A" é uma classe "B", ou
- A classe "A" é uma especialização da classe "B", ou
- A classe "B" é uma generalização da classe "A", ou
- A classe "A" é uma subclasse da classe "B", ou
- A classe "B" é a superclasse da classe "A"

Todas essas leituras podem ser utilizadas ao longo de um estudo sobre as classes de um sistema. Vale ressaltar que não existe um número limite de heranças, sendo possível criarmos o conceito de netos, bisnetos, etc..

No exemplo da figura abaixo, temos três classes sendo representadas: **Pessoa**, **Professor** e **Aluno**. A classe **Pessoa** possui os atributos *Idade*, *Nome* e *RG*. Ela é a superclasse das classes **Professor** e **Aluno**, que foram especializadas por conta dos atributos e métodos particulares existentes para cada tipo de objeto. Lembrando que a classe professor terá como atributos, além das informações de *Disciplinas* e *Matricula*, os atributos da classe pai *Idade*, *Nome* e *RG.*



Exemplo de herança

Princípio do Encapsulamento

O objetivo do princípio de Encapsulamento é ocultar dados e/ou operações existentes em uma classe. A ideia de ocultar tais características e/ou comportamentos tem forte relação com segurança do objeto. Dessa maneira, o objeto só terá disponível conteúdos realmente necessários para sua interação. Sendo assim, o encapsulamento auxilia também na independência do objeto.

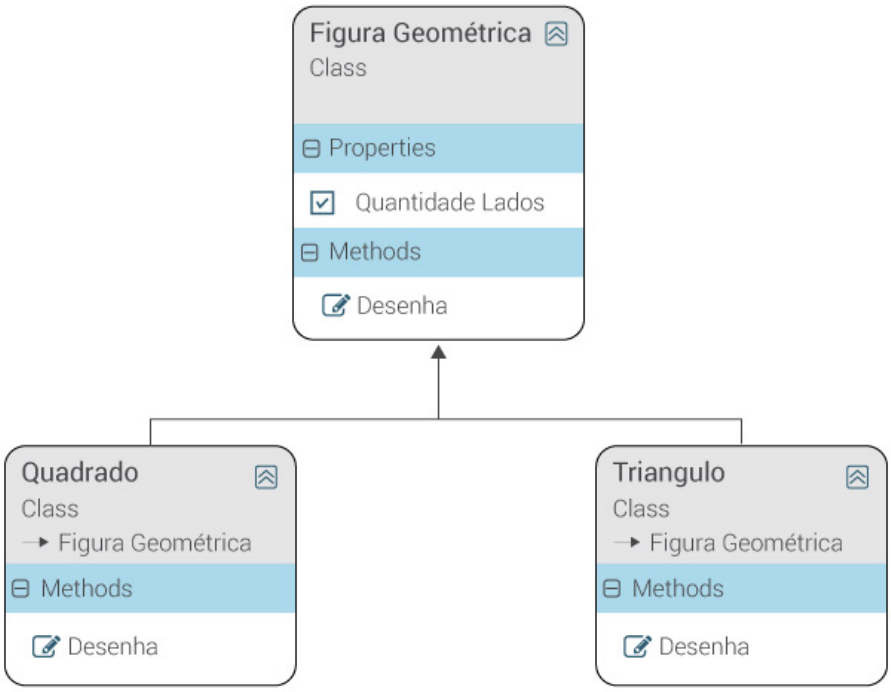
Existem diferentes níveis de encapsulamento, que serão estudados com maiores detalhes no tópico vinculado ao diagrama de classes. Esses diferentes níveis se fazem necessários por conta da arquitetura do sistema em si, que pode estar segmentado em diferentes camadas.

A definição do que pode ou não estar visível para outros objetos acessarem vai depender das definições de requisitos do sistema que está sendo modelado.

Princípio do Polimorfismo

A orientação a objetos permite que uma subclasse possa alterar o comportamento / forma como a sua superclasse executa determinada operação, mantendo o mesmo nome de operação. Esse princípio é chamado de polimorfismo, visto que essa possiblidade implica em um sistema poder ter múltiplas formas de execução de uma operação que possua a mesma assinatura (declaração do método).

O exemplo abaixo mostra a capacidade do polimorfismo. A superclasse FiguraGeometrica é apenas uma simplificação do objeto real, indicando que toda figura geométrica tem uma quantidade de lados e que toda figura geométrica tem a operação *Desenha*. Entretanto, entende-se que a operação *Desenha* irá modificar o seu comportamento de acordo com o tipo de figura geométrica que estiver sendo desenhada.



Exemplo de polimorfismo

Recapitulando

Neste tópico vimos os princípios de Herança, Encapsulamento e Polimorfismo, utilizados pela Orientação a Objetos para administrar a complexidade e facilitar a modelagem de sistemas.

Quiz

Exercício Final

Conceitos de Orientação a Objetos

INICIAR

Referências

BOOCK, Grady; JACOBSON, Ivar; RUMBAUGH, James. **UML**: guia do usuário. Rio de Janeiro: Campus, 2000.

PRESSMAN, R. S. **Engenharia de Software**: Uma abordagem profissional. 7ª. ed. Porto Alegre: AMGH, 2011.

SOMMERVILLE, I. **Engenharia de Software**. 9ª. ed. São Paulo: Pearson Prentice Hall, 2011.

Avalie este tópico

ANTERIOR
Histórico das Metodologias de Desenvolvimento de Sistemas

Índice

PRÓXIMO
Abstração: Conceitos e Exemplos

Ajuda?
(https://ava.uninove.br/seu/AVA/topico/topico.php?idCurso=)

Biblioteca (https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove (http://www.uninove.br)
Mapa do Site