

Programação Orientada a Objetos

Andre Peil, Daniel Retzlaff, Marlon Dias

24 de outubro de 2011

Programação Orientada a Objetos

Programação Orientada a objetos em Ciência da Computação é um paradigma que interage com diversas unidades de software, chamados de objetos. Dentro deste paradigma de programação existem dois conceitos que são de fundamental importância para o entendimento da manipulação dos dados com orientação a objetos: herança e polimorfismo. Onde, a partir destes, torna-se possível a reutilização de código, e a utilização de uma mesma função, porém derivando ela em outras subfunções distintas.

Herança em programação orientada a objetos é quando uma subclasse herda, ou seja, recebe atributos e métodos de sua classe superior, chamada superclasse. Este princípio é utilizado para o reaproveitamento de código. Por exemplo, o programador pode criar uma classe pessoa, onde esta classe tem determinados atributos, como nome, idade, telefone e endereço; essas características da classe pessoa podem ser as mesmas de uma classe chamada funcionário, porém com alguns dados a mais, como a identificação do funcionário na empresa, salário, e afins. A ideia de herança é que funcionário herde os atributos que necessita da classe pessoa e utilize-o junto com os outros atributos novos referentes à classe funcionário.

Superclasse / Subclasse

O principal conceito em herança é a relação entre super classes e sub-classes. A **superclasse** é uma classe superior na hierarquia de classes, é uma classe mais geral, a classe da qual as subclasses herdaram as características. Enquanto a **subclasse** é uma classe mais específica, a que herda as características da superclasse, tendo assim os métodos e atributos da superclasse, somado com os seus particulares.

Generalização / Especificação

Podemos dizer que **generalização** é o agrupamento de objetos ou elementos com características comuns em um modelo ou sistemas. É uma descrição mais geral sobre o objeto referente. Exemplo:

Temos a superclasse móvel, e temos as subclasses cadeira, mesa. Quando falamos que a cadeira e a mesa pertencem à classe móvel estamos generalizando os objetos cadeira e mesa, pois eles têm as mesmas características que compõem um móvel.

E a **especialização** é processo inverso, é a definição das particularidades de cada objeto ou elemento. São elementos mais consistentes que estendem o elemento genérico. Exemplo:

Temos as subclasses cadeira e mesa que compõem a classe móvel, porém, a cadeira tem métodos diferentes dos da mesa, como, por exemplo, sentar e a mesa, apoiar pratos. Quando fazemos estas classificações estamos especializando a classe cadeira e a classe mesa.

Polimorfismo / Ligação dinâmica

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse possam invocar métodos que têm a mesma identificação, mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. A decisão sobre qual o método que deve ser selecionado, de acordo com o tipo da classe derivada, é tomada em tempo de execução, através do mecanismo de ligação dinâmica.

Protected

Restringe a visibilidade do membro modificado, atributo ou método, de forma que classes não-relacionadas não possam acessá-lo. Objetos da própria classe, de classes derivadas desta têm acesso a membros protected.