

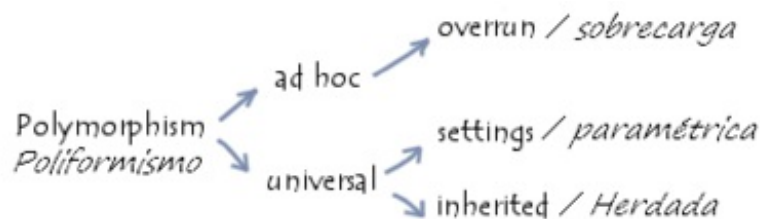
POO - O polimorfismo

Maio 2017

1. Definição do polimorfismo
2. O polimorfismo ad hoc
3. O polimorfismo paramétrico
4. O polimorfismo de inclusão

Definição do polimorfismo

A palavra polimorfismo vem do grego e significa aquilo que pode tomar várias formas. Esta característica é um dos conceitos essenciais da **programação orientada a objeto** (POO). Enquanto a herança se refere às classes e a sua hierarquia, o polimorfismo diz respeito aos métodos dos objetos. Existem três tipos de polimorfismo: **ad hoc**, que significa sobrecarga (overloading), **paramétrico**, que quer dizer modelo com uma estrutura predefinida (template) e **polimorfismo de herança**, ou seja, de redefinição e especialização (overriding). Veja em imagem os tipos de polimorfismo:



Vamos agora tentar entender mais precisamente tudo isto, mas saiba que existe muita confusão quando se trata de diferenciar todos os tipos de polimorfismo.

O polimorfismo ad hoc

O **polimorfismo ad hoc** permite ter funções com o mesmo nome, com funcionalidades similares, em classes sem nenhuma relação entre elas (a não ser, claro, serem originárias da classe objeto). Por exemplo, a classe complexa, a classe imagem e a classe link podem ter, cada uma, uma função **Exibir**. Desta forma, você não precisará se preocupar com o tipo do objeto a ser mostrado na tela. O polimorfismo ad hoc permite definir operadores cuja utilização será diferente de acordo com o tipo de configurações que lhes são próprias. Por isso, é possível sobrecarregar o operador e fazê-lo realizar ações diferentes para cada uma das operações entre dois inteiros (adição) ou entre duas cadeias de caracteres (concatenação).

O polimorfismo paramétrico

O **polimorfismo paramétrico** representa a possibilidade de definir várias funções do mesmo nome, porém

com parâmetros diferentes (em número e/ou tipo). O polimorfismo paramétrico torna possível a escolha automática do método a ser adotado em função do tipo de dado passado em parâmetro. Assim, podemos definir vários métodos homônimos **addition()** efetuando uma soma de valores: o método **int addition** (int, int) poderá retornar a soma de dois inteiros; o método **float addition** (float, float) poderá retornar a soma de dois pontos flutuantes; o método **char addition** (char, char) poderá definir, a critério do autor, a soma de dois caracteres; etc. Chamamos **assinatura** o número e o tipo dos argumentos de uma função. Assim sendo, é a assinatura de um método que determina qual será chamada.

O polimorfismo de inclusão

A possibilidade de redefinir um método em classes herdeiras de uma classe básica é chamada de **especialização**. Assim sendo, é possível chamar o método de um objeto sem se preocupar com o seu tipo intrínseco de polimorfismo de inclusão. Isso permite fazer abstração dos detalhes das classes especializadas de uma família de objeto, ocultando-os através de uma interface comum, chamada de classe básica. Imaginemos um jogo de xadrez com o rei, a rainha, o bispo, o cavalo, a torre e o peão que herdaram, cada um deles, o objeto **piece** (peça). O método **movement** (movimento) poderia, graças ao polimorfismo de inclusão, efetuar o movimento correspondente em função da classe do objeto chamado. Isto permite que o programa execute o **piece.movement** (movimento de peça) sem ter que se preocupar com a classe de cada peça.

[◀ Anterior](#)

- [1](#)
- [2](#)
- [3](#)
- [4](#)

Este documento, intitulado '[POO - O polimorfismo](#)', está disponível sob a licença [Creative Commons](#). Você pode copiar e/ou modificar o conteúdo desta página com base nas condições estipuladas pela licença. Não se esqueça de creditar o [CCM \(br.ccm.net\)](#) ao utilizar este artigo.