

Para entendermos o que é polimorfismo, precisamos revisar alguns conceitos:

- . O que são 'coisas' abstratas e concretas ?
- . Como funciona o mecanismo de Herança ?
- . Concretas: Conjunto das coisas do mundo onde vivemos. Ex: Gato e Cachorro.
- . Abstrato: Conceitos, características e comportamentos que associamos a certas 'coisas'. Ex: Mamífero.

A partir desta ideia, é possível construir as classes abstratas/interfaces do sistema.

Quais são as características de um cliente e como ele se comporta no sistema ?

Quais são as características desse pagamento e como ele se comporta dentro do sistema ?

Imagine que estamos desenvolvendo um sistema de locação onde o cliente pode alugar um carro, barco ou helicóptero.

Se resolvêssemos criar uma classe para cada 'veículo', teríamos que escrever um código de locação para cada 'veículo'. Se observarmos, vemos que cada veículo pertence a uma classe concreta, mas estão conectadas a ideia de um 'veículo' que seria a classe abstrata. A partir desta ideia, podemos criar um único código de locação preparado para funcionar recebendo um tipo de veículo e a partir do veículo podemos derivar outras classes.

Algumas características/comportamentos serão importantes no processo de locação e iremos amarrá-los na classe 'veículo'. Exemplo: Todo veículo terá uma identificação, a quantidade de combustível na hora da locação e a quantidade de quilômetros rodados até o momento da locação.

Na herança, todas as características das classes abstratas são transportadas para as classes concretas. (reutilização de código).

Polimorfismo:

Imagine que estamos criando uma classe 'pagamento'(abstrata).

- . Quais são as características e comportamentos dessa classe ?

Como o nome diz, todo pagamento paga alguma coisa. Mas como será efetuado esse pagamento ? Por boleto ? cartão de crédito ? ou dinheiro ?

1º passo - Criamos uma classe abstrata chamada 'pagamento'

2º passo - Colocamos o método 'pagar' dentro da classe 'pagamento' - Não terá nenhum algoritmo ligado ao método 'pagar' uma vez que ainda não sabemos como o pagamento será efetuado.

3º passo - criamos as classes concretas 'pagamentoDinheiro', 'pagamentoCartao' e 'pagamentoBoleto'. Cada classe concreta terá um método 'pagar' com um algoritmo específico.

A partir deste momento já temos uma ideia de como funciona o polimorfismo.

Se codificarmos o sistema de locação preparado para receber a classe 'pagamento' e invocar nessa classe o método 'pagar', através do polimorfismo, conseguimos fazer com que o pagamento seja feito utilizando o algoritmo de boleto, dinheiro ou cartão de crédito. Só dependerá da forma como instanciamos o objeto de pagamento.

Neste exemplo, o polimorfismo se dá quando o objeto 'pagamento' se comporta como 'pagamentoDinheiro', 'pagamentoCartao' ou 'pagamentoBoleto' dependendo da forma como ele foi instanciado.