Departamento de Ciência da Computação ? IME-USP

MAC 413 ? Tópicos de Programação Orientada a Objeto
Prof.: Fabio Kon

PADRÃO DE PROJETO

Grupo:
Maíra de Assis Ramos (mramos@linux.ime.usp.br) 3286231
Marcus M. S. Moysés (mmoyses@linux.ime.usp.br) 3286415

MANIPULADOR DE DADOS

Intenção

Fazer com que operações de dados únicas possam ser utilizadas em conjuntos de tipos de dados diferentes.

Motivação

Algumas vezes, uma operação única deve ser realizada para conjuntos diferentes de dados e/ou diferentes configurações.

Considere, por exemplo, um conjunto de cadastros que devem realizar buscas, inclusões, exclusões e alterações. O cadastro de clientes deve manipular o nome (texto, no máximo 50 caracteres) e a idade (inteiro positivo menor que 120). O cadastro de produto deve manipular um código (inteiro positivo), uma descrição (texto qualquer) e um preço (valor monetário). Como implementar todas as operações para cada um dos cadastros?

Uma solução seria criar um cadastro com toda a funcionalidade e simplesmente utilizá-lo como modelo, copiando-o e fazendo as alterações necessárias. Isso não está longe do que ocorre na realidade.

Uma outra saída é criar a funcionalidade em operações distintas, modularizadas recebendo uma classe que implementa as configurações necessárias para cada operação realizar sua tarefa. Então, para cada operação (busca, inclusão, etc) cria-se uma classe que a implementa, uma interface que a classe de configuração deve respeitar e uma interface que deve ser respeitada pelos tipos de dados que a classe de configuração manipula. É definida ainda uma interface para a classe manipuladora definindo qual funcionalidade deve ser oferecida.

No exemplo, as classes operadoras são Busca, Inclusão, Exclusão e Alteração. As classes manipuladoras (configuradoras) são CadastroCliente e CadastroProduto. A classe CadastroCliente implementa uma interface para cada operação e a interface para a funcionalidade Cadastro: buscar(), incluir(), excluir(), alterar(). Ela manipula os tipos nome e idade que também devem implementar cada interface definida para as operações.

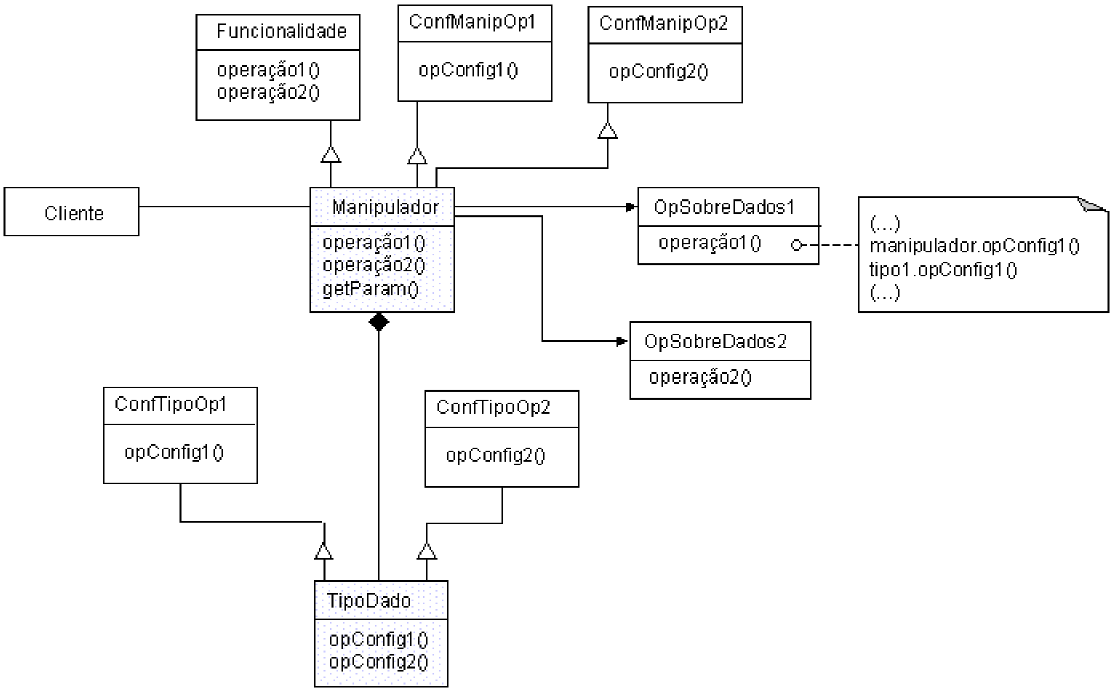
Dessa forma as operações são implementadas apenas uma vez, de maneira modularizada e independente dos dados sobre os quais operam.

Aplicabilidade

Use o *padrão* Manipulador de Dados quando:

- você tem uma ou mais operações que se repetem e variam apenas de acordo com os dados manipulados;

Estrutura



Participantes

- **Funcionalidade** (Cadastro)
- Define a interface específica que o Manipulador deve implementar para oferecer a funcionalidade que o cliente espera.
- **Manipulador** (CadastroCliente, CadastroProduto)
- Implementa a interface Funcionalidade e delega suas operações à classe que opera sobre os dados.
- Implementa a configuração para Operação e agrega os tipos de dados que devem ser manipulados.
- **ConfManipOp** (ConfCadBusca, ConCadfInclusão, etc)
- Define a interface de configuração para o Manipulador necessária para que a Operação realize suas tarefas.
- **Operação** (Busca, Inclusão, etc)
- Realiza uma operação sobre os dados manipulados pelo Manipulador, utilizando interfaces de configuração específica.
- **TipoDado** (nome, idade, preço, etc)
- Implementa ConfTipoOp para que a operação seja realizada.
- **ConfManipOp** (ConfTipoBusca, ConfTipoInclusão, etc)
- Define a interface de configuração para o TipoDado necessária para que a Operação realize suas tarefas.

Colaborações

- Manipulador repassa solicitações dos clientes para o seu objeto Operação

Conseqüências

O *padrão* Manipulador de dados tem as seguintes conseqüências:

1. Desacopla a implementação da funcionalidade dos tipos de dados manipulados.
2. Permite que as operações sejam implementadas apenas uma vez e reaproveitadas.
3. Aumenta a complexidade do projeto.

Implementação

1. *Implementação das interfaces da classe Manipulador.* Nas linguagens onde não é possível implementar várias interfaces como C++ essa solução pode implicar herança múltipla.
2. *Instanciação da classe Manipulador.* Como garantir que a classe Manipulador será instanciada corretamente com os tipos de dados? Uma solução é utilizar o padrão Abstract Factory.