

Programação Orientada a Objetos II

Prof. Walter Gima



Objetivo

Prover isolamento da tecnologia de persistência.

Propósito

O padrão Data Access Object (DAO) é um padrão introduzido no ambiente JEE ^[3] para simplificar e desacoplar a interação das aplicações Java com a API JDBC.

O Problema

A grande maioria das aplicações de nível corporativo usam algum tipo de persistência de dados. Entre eles o mais usado é o Banco de Dados. A linguagem SQL é amplamente utilizada para comunicar com os sistemas gerenciadores de banco de dados (SGBD). Java suporta esta necessidade desde cedo através da API JDBC (Java Database Connectivity).

Antes do uso de Java em sistemas corporativos as aplicações eram majoritariamente escritas em linguagens orientadas a processo e a ordem e instruções SQL específicas continham regras de negócio. Ao passar esses sistemas para Java, essas regras não se poderiam perder e, ao mesmo tempo, há que se trabalhar com objetos. O padrão DAO visava originalmente encapsular esses conjuntos de códigos SQL que existiam em aplicações legadas [3]. Esse código continha regras tanto na pesquisa dos dados, tanto na edição dos dados. É comum ao inserir um determinado dado, ter que inserir ou atualizar outros.

O problema que o padrão DAO tenta resolver tem, na realidade, três perspectivas:

- Legado : Queremos encapsular lógicas de persistência legadas escritas com SQL ou outras tecnologias de forma simples. Queremos apenas utilizar as tecnologias Java, mas manter as mesmas regras de negócio.
- Isolamento : Queremos que a aplicação seja isolada da API com que está comunicando. Queremos poder alterar a API utilizada internamente sem alterar o contrato de acesso aos dados. No caso específico do JDBC, queremos ainda poder utilizar diferentes dialetos do SQL conforme o SGBD que está sendo utilizado para tirar o máximo de proveito das especificidades de cada SGBD.
- Mapeamento de Objetos : Queremos utilizar objetos no ambiente Java. Queremos poder utilizar os mesmos objetos independentemente da API de persistência. Normalmente isto passa por respeitar algum Mapeamento Objeto-Relacionamento ORM (Object-Relational Mapping). Isto pode ser feito de forma automatizada ou não. Outros mapeamentos são possíveis, dependendo da tecnologia subjacente; como para XML, LDAP ou WebServices. Os mapeamentos ORM podem ainda mudar quando mudamos de gerenciador de banco de dados

A solução

O objeto DAO é responsável por operar o mecanismo de persistência em nome da aplicação tipicamente executando os quatro tipos de operações - Criar, Recuperar, Alterar, Apagar - conhecidas pela sigla CRUD - do inglês Create, Retrieve, Update, Delete. As operações de edição são invocadas diretamente passando o objeto com as informações a serem editadas. As operações de recuperação são normalmente implementadas como métodos específicos. Por exemplo, recuperar o objeto que corresponde com uma certa chave, ou critério de busca. Estes métodos contêm regras de negócio diferentes conforme o tipo de dados sendo persistido. Em particular pode não existir nenhuma regra particular, ou a regra só poder ser executada em um tipo específico de API de persistência.

O DAO é um tipo muito específico de Serviço (padrão Service) especializado em conversar com o SGBD e em prover as operações CRUD e de pesquisas. Por isto o contrato de um DAO é diferenciado da sua implementação (como é normal no padrão Service) e cada implementação corresponde a uma tecnologia específica e/ou a um SGBD específico. Em última análise não se trata de uma única implementação, mas de uma família de implementações. Cada família especificamente desenhada para um tecnologia de persistência, um SGBD e até um conjunto específico de objetos de transporte.

[view plain](#) [copy to clipboard](#) [print](#) ?

```
01. public Class ModelCliente{
02.     private int codigo;
03.     private String nome;
04.
05.     public int setCodigo(int codigo)...
06.     public int getCodigo()...
07.
08.     public String setNome(String nome)...
09.     public String getNome()...
10. }
```

Camada DAO

```
view plain copy to clipboard print ?  
01. public Class DaoCliente{  
02.  
03.     public boolean inserir..  
04.     public boolean deletar...  
05.     public boolean editar...  
06.     ...  
07. }
```

Camada CONTROLLER (APENAS PRESENTE NO MODELO MVC)

```
view plain copy to clipboard print ?
01. public Class ControllerCliente{
02.     public boolean InserirCliente(){
03.         mCliente = new ModelCliente();
04.         mCliente.setCodigo(1);
05.         mCliente.setNome('Ronaldo');
06.
07.         daoCliente = new DaoCliente();
08.         daoCliente.inserir(mCliente);
09.     }
10.     ....
11. }
```

```
ContatoDAO contatoDao =  
new ContatoMySqlDAO();  
new ContatoOracleDAO();  
new ContatoSQLDAO();
```



Anhanguera

Dúvidas ?