

Padrões de Projeto

Active Record

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Rio Grande
Divisão de Computação

Agenda

- 1 Introdução
 - Aplicabilidade
- 2 Implementação
- 3 Analisando o Padrão

Introdução

A parte de acesso ao banco de dados é uma das mais interessantes

Ela é responsável por se conectar ao banco de dados e extrair, inserir e atualizar as informações.

- É responsável também por transformar modelos de Objetos em modelos relacionais, o tal do ORM (Mapeamento Objeto Relacional), já que em muitos casos lidamos com banco de dados relacionais.

Introdução

Podemos acessar de 2 formas o banco de dados:

- ➊ Usando um framework
- ➋ Escrevendo SQL próprio.

Hoje em dia existe uma variedade de frameworks, que programam os vários recursos necessários, de modo que costuma ser perda de tempo fazer o acesso direto ao Banco de Dados.

- Geralmente o ORM é feito através desses frameworks.

Temos dois padrões comuns para a comunicação com o BD/Framework:

- o Data Access Objet (DAO)
- o Active Record (AR)

Introdução

Definição

Em Engenharia de software, active record é um padrão de projeto encontrado em softwares que armazenam seus dados em Banco de dados relacionais.

- Assim foi nomeado por Martin Fowler em seu livro *Patterns of Enterprise Application Architecture*.

Característica Básica:

A interface de um certo objeto deve incluir funções como por exemplo:

- Inserir (*Insert*), Atualizar (*Update*), Apagar (*Delete*)
- E propriedades que correspondam de certa forma diretamente às colunas do banco de dados associado.

Introdução

Um pouco mais de teoria....

Active record é uma abordagem para acesso de dados num banco de dados.

Uma tabela de banco de dados ou visão (view) é embrulhada (wrapped) em uma classe.

- Portanto, uma instância de um objeto é amarrada a um único registro (tupla) na tabela.

Introdução

Exemplo:

- Após a criação e gravação de um objeto, um novo registro é adicionado à tabela.
- Qualquer objeto carregado obtém suas informações a partir do banco de dados.
- Quando um objeto é atualizado, o registro correspondente na tabela também é atualizado.

A classe de embrulho implementa os métodos de acesso (setter e getter) ou propriedades para cada coluna na tabela ou visão.

Aplicabilidade

Este padrão é comumente utilizado por ferramentas de persistência de objetos e em mapeamento objeto-relacional.

- **ActiveJDBC**
 - `http://javalite.io/activejdbc`
- E etc...

Implementação

Implementações do conceito podem ser encontradas em vários Frameworks para diversos ambientes de programação.

- Por exemplo, se um banco de dados possui a tabela **produtos** com as colunas **nome** (tipo *string*) e **valor** (tipo *number*) o padrão de projeto Active Record é implementado na classe Produto com o seguinte pseudo-código:

```
produto = new Produto()  
produto.nome = "Produto exemplo"  
produto.valor = 123.45  
produto.save()
```

Figure: Exemplo de Utilização - Padrão Active Record.

Irá criar um novo registro na table produtos com os valores fornecidos sendo grosseiramente equivalente ao comando SQL:

```
INSERT INTO produtos (nome, valor) VALUES ('Produto exemplo', 123.45);
```

Implementação

Da mesma forma, a classe pode ser usada para consultar o banco de dados:

```
b = Produto.find_first("nome", "televisor")
```

Este código criará um novo objeto do tipo **Produto** baseado no primeiro registro encontrado da tabela **produtos** onde a coluna **nome** contém o valor "televisor".

- O comando SQL usado pode ser similar ao seguinte (dependendo dos detalhes da implementação SQL do banco de dados):

```
SELECT * FROM produtos WHERE nome = 'televisor' LIMIT 1; -- MySQL ou PostgreSQL
```

Analisando o Padrão

Analisando a descrição do padrão, percebemos claramente que existem duas responsabilidades no mesmo objeto: persistência e lógica de domínio. Com uma implementação mal feita, este padrão poderá violar o **Princípio da Responsabilidade Única**.

Essa abordagem do Active Record é muitas vezes considerada uma falha no design da aplicação, pelo fato de que o Domínio passa a ser subordinado do Banco de dados.

Este argumento pode ser rebatido já que, na maioria dos sistemas, as classes do Modelo são apenas representações do Domínio, e não programam regras em si, e que essas regras passam estar representadas na persistência, não afetando o design de muitas aplicações.

Padrões de Projeto

Active Record

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Rio Grande
Divisão de Computação