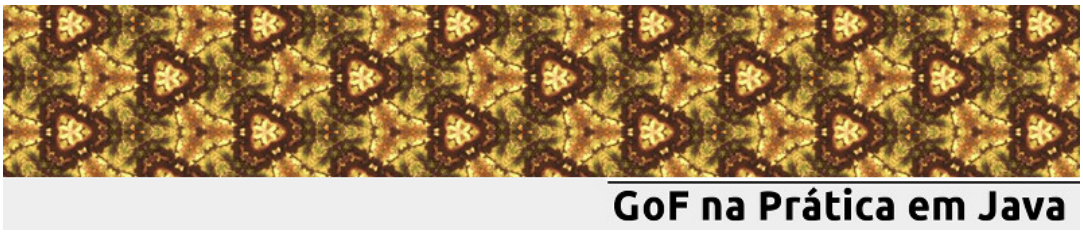
Proxy

Fernando Anselmo
GoF na Prática em Java

Função deste Padrão

Prover um substituto ou ponto, através do qual um objeto possa controlar o acesso de outro.



1 Ficha do Padrão

Tipo : Estrutural, diz respeito como classes e objetos podem ser combinados para formar grandes estruturas.

Conhecimentos : Interface e Classes Concretas.

Consequências : Introduz um nível de intermediação para acessar um objeto. Essas intermediações adicionais possuem diversos usos tais como: esconder o fato de que um objeto reside em um espaço de endereçamento diferente, criação de objetos sob demanda e permite que sejam realizadas tarefas adicionais quando um objeto é acessado.

É usado quando : Para prover um representante local para um objeto em um espaço de endereçamento diferente. Controlar o uso do objeto original, o que é muito útil quando os clientes devem ter diferentes permissões de acesso ao objeto.

2 Problema

Um determinado sistema precisa acessar um objeto SerSupremo porém este objeto não está disponível (por ser remoto).

3 Prévia Estrutura de Classes

Interface com a estrutura para acesso ao objeto concreto:

Listagem 1: *Interface SerSupremo*

```
1 interface SerSupremo {
2   public String responder(String pergunta);
3 }
```

Classe concreta que deve ser acessada pelo usuário:

Listagem 2: *Classe SerSupremoReal*

```
class SerSupremoReal implements SerSupremo {
  public String responder(String pergunta) {
    return "Porque 42 é a resposta para tudo";
  }
}
```

4 Aplicação do Padrão

Classe de intermediação com o SerSupremoReal:

Listagem 3: *Classe Intermediario*

```
class Intermediario implements SerSupremo {
  private SerSupremo contato;

  public Intermediario() {
    contato = new SerSupremoReal();
  }
  public String responder(String pergunta) {
    return contato.responder(pergunta);
  }
}
```

Classe com um exemplo de uso pelo cliente:

Listagem 4: *Classe Usuario*

```
public class Usuario {
  public static void main(String [] args) {
    new Usuario().problema();
  }
  public void problema() {
    SerSupremo intermediario = new Intermediario();
    System.out.println(intermediario.responder("Qual o sentido da Vida?"));
  }
}
```

Referências

[1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides *Design Patterns. Elements of Reusable Object-Oriented Software 1 ed.* Estados Unidos, Addison-Wesley, 1995, ISBN 0-201-63361-2.