

Façade

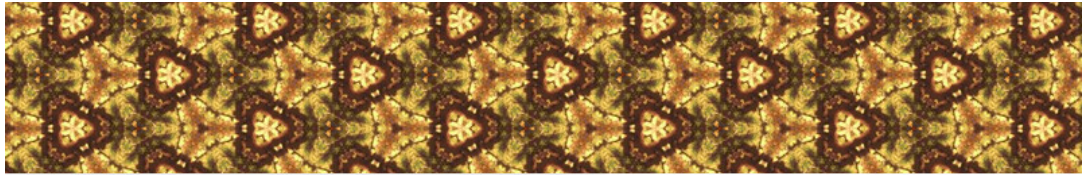
Fernando Anselmo

GoF na Prática em Java

Função deste Padrão



Oferecer uma interface única para um conjunto de interfaces de um subsistema. Define uma interface de nível mais elevado que torna o sistema mais fácil de usar.



GoF na Prática em Java

1 Ficha do Padrão

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

Conhecimentos : Enumeração e Classes Concretas.

Consequências : Isola os clientes dos componentes do subsistema e reduz o número de objetos com que o cliente interage. Permite que o subsistema seja muito mais fácil de se usar. Ajuda a estruturar em camadas não só do sistema, bem como suas dependências entre os objetos.

É usado quando : Existem muitas dependências entre os clientes e as classes de implementação de uma abstração, introduzir o Façade causa o desacoplamento do subsistema dos clientes e dos outros subsistemas. Se deseja oferecer uma interface simples para um subsistema complexo.

2 Problema

Uma loja de Discos vende e aluga CDs com base em uma determinada categoria, cada venda gera 50% e aluguel 5% do valor para a Gravadora, ao final do dia deve ser tirado um relatório quantos aluguéis ou vendas foram realizadas e o valor de repasse a gravadora.

3 Prévia Estrutura de Classes

Lista de enumeração com as possíveis categorias:

Listagem 1: Enumeração Categoria

```
1 enum Categoria {  
2     PROMOCIONAL,  
3     LANCAMENTO,  
4     NORMAL  
5 }
```

Classe base para os CDs a serem vendidos ou alugados:

Listagem 2: Classe CD

```
1 class CD {
2     private float preco;
3
4     public CD(Categoria categoria) {
5         switch (categoria) {
6             case PROMOCIONAL: preco = 9.99f;
7             case LANÇAMENTO: preco = 29.99f;
8             case NORMAL: preco = 19.99f;
9         }
10    }
11    public float getPreco() {
12        return preco;
13    }
14 }
```

Classe que exemplifica o cliente com seu saldo:

Listagem 3: Classe Cliente

```
1 class Cliente {
2     private short comprar;
3     private short alugar;
4
5     public void addComprar() {
6         this.comprar++;
7     }
8     public void addAlugar() {
9         this.alugar++;
10    }
11    public String toString() {
12        return "[Cliente] compras: " + this.comprar + " aluguel: " + this.alugar;
13    }
14 }
```

Classe que exemplifica a Gravadora que obterá o repasse:

Listagem 4: Classe Gravadora

```
1 class Gravadora {
2     private float repasse;
3
4     public void setRepasse(float repasse) {
5         this.repasse = repasse;
6     }
7     public String toString() {
8         return "[Gravadora] Repasse: " + this.repasse;
9     }
10 }
```

4 Aplicação do Padrão

Classe façade das Regras de Negócio:

Listagem 5: Classe RNFacade

```
1 class RNFacade {
2     private Cliente cliente;
3     private Gravadora gravadora;
```

```

4
5 public RNFacade() {
6     cliente = new Cliente();
7     gravadora = new Gravadora();
8 }
9 public void comprar(Categoria categoria) {
10     CD cd = new CD(categoria);
11     cliente.addComprar();
12     gravadora.setRepasse(cd.getPreco() * 0.5f);
13 }
14 public void alugar(Categoria categoria) {
15     CD cd = new CD(categoria);
16     cliente.addAlugar();
17     gravadora.setRepasse(cd.getPreco() * 0.05f);
18 }
19 public String toString() {
20     return cliente + " " + gravadora;
21 }
22 }

```

Classe com um exemplo de uso pelo cliente com algumas transações:

Listagem 6: *Classe LojaDisco*

```

1 public class LojaDisco {
2     private RNFacade facade = new RNFacade();
3
4     public static void main(String [] args) {
5         new LojaDisco().trabalhar();
6     }
7     public void trabalhar() {
8         facade.comprar(Categoria.NORMAL);
9         facade.alugar(Categoria.LANCAMENTO);
10        facade.comprar(Categoria.PROMOCIONAL);
11        System.out.println(facade);
12    }
13 }

```

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.