

Decorator

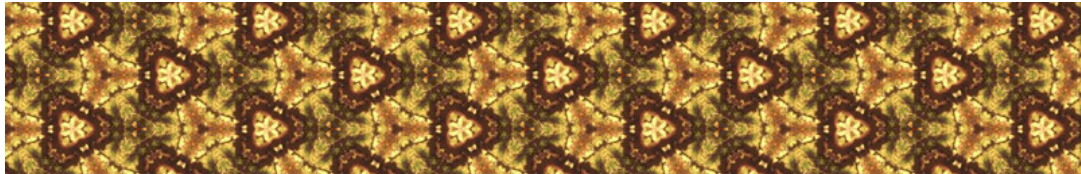
Fernando Anselmo

GoF na Prática em Java

Função deste Padrão

A

nexar responsabilidades adicionais a um objeto dinamicamente. Decorators oferecem uma alternativa flexível ao uso de herança para entender uma funcionalidade.



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 : Interfaces, Classes Abstratas e Classes Concretas.

Consequências : Maior flexibilidade do que herança estática para evitar a sobrecarga de características em classes de alto nível da hierarquia. Um Decorator e seu componente são idênticos. Multidão de pequenos objetos: fácil de configurar porém difícil para compreender.

É usado quando : adicionar responsabilidades a objetos individuais de forma dinâmica e transparente, sem afetar outros objetos. Retirar responsabilidade de extensões através de subclasses que são impraticáveis.

2 Problema

Com base na interface Ervilha o usuário deseja criar combinações de várias cores e isso deve ser realizado de modo simplificado.

3 Prévia Estrutura de Classes

Interface com a estrutura para criação das ervilhas:

Listagem 1: *Interface Ervilha*

```
1 interface Ervilha {  
2     void combinar();  
3     void setErvilhaPai(Ervilha ervilhaPai);  
4 }
```

4 Aplicação do Padrão

Classe abstrata para estrutura do padrão de decoração:

Listagem 2: *Classe Abstrata ErvilhaDecorator*

```
1 abstract class ErvilhaDecorator implements Ervilha {
2     private Ervilha ervilhaPai = null;
3
4     public ErvilhaDecorator(Ervilha ervilhaPai) {
5         this.ervilhaPai = ervilhaPai;
6     }
7     public void setErvilhaPai(Ervilha ervilhaPai) {
8         this.ervilhaPai = ervilhaPai;
9     }
10    public void combinar() {
11        if (ervilhaPai != null) {
12            ervilhaPai.combinar();
13            System.out.print(" x ");
14        } else {
15            System.out.println();
16        }
17    }
18 }
```

Classe que implementa a ervilha tipo Amarela:

Listagem 3: *Classe ErvilhaAmarela*

```
1 class ErvilhaAmarela extends ErvilhaDecorator {
2     public ErvilhaAmarela(Ervilha ervilhaPai) {
3         super(ervilhaPai);
4     }
5     public ErvilhaAmarela() {
6         super(null);
7     }
8     public void combinar() {
9         super.combinar();
10        mostrarCor();
11    }
12    private void mostrarCor() {
13        System.out.print("Ervilha Amarela");
14    }
15 }
```

Classe que implementa a ervilha tipo Verde:

Listagem 4: *Classe ErvilhaVerde*

```
1 class ErvilhaVerde extends ErvilhaDecorator {
2     public ErvilhaVerde(Ervilha ervilhaPai) {
3         super(ervilhaPai);
4     }
5     public ErvilhaVerde() {
6         super(null);
7     }
8     public void combinar() {
9         super.combinar();
10        mostrarCor();
11    }
12    private void mostrarCor() {
```

```
13     System.out.print("Ervilha Verde");
14 }
15 }
```

Classe que implementa a ervilha tipo Vermelha:

Listagem 5: *Classe ErvilhaVermelha*

```
1 class ErvilhaVermelha extends ErvilhaDecorator {
2     public ErvilhaVermelha(Ervilha ervilhaPai) {
3         super(ervilhaPai);
4     }
5     public ErvilhaVermelha() {
6         super(null);
7     }
8     public void combinar() {
9         super.combinar();
10        mostrarCor();
11    }
12    private void mostrarCor() {
13        System.out.print("Ervilha Vermelha");
14    }
15 }
```

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

Listagem 6: *Classe Geneticista*

```
1 public class Geneticista {
2     public static void main(String [] args) {
3         new Geneticista().combine();
4     }
5     public void combine() {
6         new ErvilhaVerde(new ErvilhaVermelha()).combinar();
7         new ErvilhaVerde().combinar();
8         new ErvilhaVerde(new ErvilhaVermelha(new ErvilhaAmarela())).combinar();
9     }
10 }
```

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.