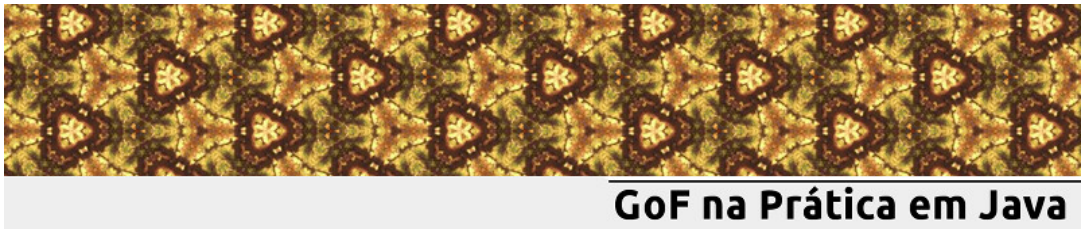
Decorator

Fernando Anselmo
GoF na Prática em Java

Função deste Padrão

Usar compartilhamento para suportar de forma mais eficiente grandes quantidades de objetos refinados.



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 : Introduz custos em tempo de execução associados a transferência e a procura dos estados extrínsecos, especialmente os que forem formalmente armazenados como estados intrínsecos. Reduz o número total de instâncias que se obtém com o compartilhamento, a quantidade de estado intrínseco por objeto.

É usado quando : Uma aplicação utiliza um grande número de objetos e os custos de armazenamento são grandes por causa dessa grande quantidade. A maior parte do estado dos objetos podem se tornar extrínsecos e a aplicação não depende da identidade dos objeto.

2 Problema

Com base na interface Ervilha o usuário deseja criar combinações de suas cores 2 a 2, sem ter que se preocupar em gerar diversos objetos.

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 }
```

Classe abstrata para estrutura de combinação:

Listagem 2: *Classe Abstrata ErvilhaCombinada*

```
abstract class ErvilhaCombinada implements Ervilha {
  private Ervilha ervilhaPai = null;

  public ErvilhaCombinada(Ervilha ervilhaPai) {
    this.ervilhaPai = ervilhaPai;
  }
  public void setErvilhaPai(Ervilha ervilhaPai) {
    this.ervilhaPai = ervilhaPai;
  }
  public void combinar() {
    if (ervilhaPai != null) {
      ervilhaPai.combinar();
      System.out.print(" x ");
    } else {
      System.out.println();
    }
  }
}
```

Classe que implementa a ervilha tipo Amarela:

Listagem 3: *Classe ErvilhaAmarela*

```
class ErvilhaAmarela extends ErvilhaCombinada {
  public ErvilhaAmarela(Ervilha ervilhaPai) {
    super(ervilhaPai);
  }
  public ErvilhaAmarela() {
    super(null);
  }
  public void combinar() {
    super.combinar();
    mostrarCor();
  }
  private void mostrarCor() {
    System.out.print("Ervilha Amarela");
  }
}
```

Classe que implementa a ervilha tipo Verde:

Listagem 4: *Classe ErvilhaVerde*

```
class ErvilhaVerde extends ErvilhaCombinada {
  public ErvilhaVerde(Ervilha ervilhaPai) {
    super(ervilhaPai);
  }
  public ErvilhaVerde() {
    super(null);
  }
  public void combinar() {
    super.combinar();
    mostrarCor();
  }
  private void mostrarCor() {
    System.out.print("Ervilha Verde");
  }
}
```

Classe que implementa a ervilha tipo Vermelha:

Listagem 5: *Classe ErvilhaVermelha*

```
class ErvilhaVermelha extends ErvilhaCombinada {
  public ErvilhaVermelha(Ervilha ervilhaPai) {
    super(ervilhaPai);
  }
  public ErvilhaVermelha() {
    super(null);
  }
  public void combinar() {
    super.combinar();
    mostrarCor();
  }
  private void mostrarCor() {
    System.out.print("Ervilha Vermelha");
  }
}
```

4 Aplicação do Padrão

Classe que implementa o conjunto das possibilidades de combinação:

Listagem 6: *Classe ErvilhaFlyWeight*

```
class ErvilhaFlyWeight {
  private Ervilha [] pool = {
    new ErvilhaVerde(),
    new ErvilhaVermelha(),
    new ErvilhaAmarela()
  };
  public void mostrar(Ervilha ervilhaPai) {
    for (int i = 0; i < pool.length; i++) {
      ervilhaPai.setErvilhaPai(pool[i]);
      ervilhaPai.combinar();
    }
  }
}
```

Classe com um exemplo de uso pelo cliente com as possíveis combinações 2 a 2:

Listagem 7: *Classe Geneticista*

```
public class Geneticista {
  public static void main(String [] args) {
    new Geneticista().combine();
  }
  public void combine() {
    ErvilhaFlyWeight fly = new ErvilhaFlyWeight();
    fly.mostrar(new ErvilhaVerde());
    fly.mostrar(new ErvilhaVermelha());
    fly.mostrar(new ErvilhaAmarela());
  }
}
```

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.