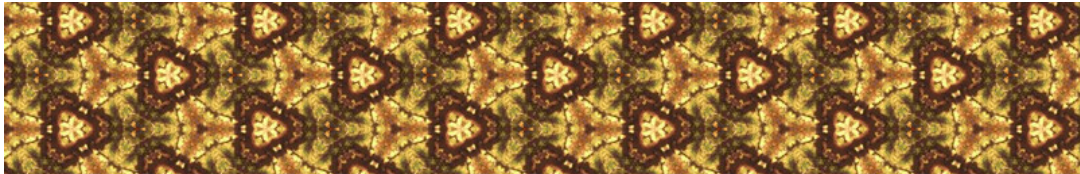

Builder

Fernando Anselmo

GoF na Prática em Java

Função deste Padrão

Separa a construção de um objeto complexo de sua representação, de modo que o mesmo processo de construção possa criar diferentes representações.



GoF na Prática em Java

1 Ficha do Padrão

Tipo : Criacional, diz respeito ao processo de criação dos objetos.

Conhecimentos : Interface, Classes Abstratas e Classes Concretas.

Consequências : Possibilita variar a representação interna de um produto; isola o código de construção e representação aumentando a modularidade; possibilita grande controle no processo de construção.

É usado quando : O algoritmo para a criação de um objeto complexo deve ser independente das partes que o compõem e de como estas são conectadas entre si; O processo de construção deve permitir a criação de diferentes representações do objeto construído.

2 Problema

O cliente, necessita criar modelos de veículos, inicialmente com carros tipo Esportivo como Ferrari e utilitários como Jeep. Este repassará as informações do Modelo para a Montadora que ordenará a construção e devolverá o Veículo com as informações necessárias.

3 Prévia Estrutura de Classes

Classe abstrata para organizar a família dos Veículos:

Listagem 1: *Classe Abstrata Veiculo*

```
1 abstract class Veiculo {
2     private String modelo;
3     private String tipo;
4     public Veiculo(String modelo, String tipo) {
5         this.modelo = modelo;
6         this.tipo = tipo;
7     }
8     protected String getModelo() {
9         return modelo;
10    }
11    protected String getTipo() {
```

```
12     return tipo;
13 }
14 }
```

Classe exemplo de veículo do modelo Ferrari:

Listagem 2: *Classe Ferrari*

```
1 class Ferrari extends Veiculo {
2     private String nome;
3     public Ferrari(String modelo) {
4         super(modelo, "Esportivo");
5         this.nome = "Ferrari";
6     }
7     public String toString() {
8         return "Carro: " + nome + " " + getModelo() + " " + getTipo();
9     }
10 }
```

Classe exemplo de veículo do modelo Jeep:

Listagem 3: *Classe Jeep*

```
1 class Jeep extends Veiculo {
2     private String nome;
3     public Jeep(String modelo) {
4         super(modelo, "Utilitario");
5         this.nome = "Jeep";
6     }
7     public String toString() {
8         return "Carro: " + nome + " " + getModelo() + " " + getTipo();
9     }
10 }
```

4 Aplicação do Padrão

Interface com a estrutura para a montadora:

Listagem 4: *interface MontadoraBuilder*

```
1 interface MontadoraBuilder {
2     void montar(char tipo, String modelo);
3     Veiculo getVeiculo();
4 }
```

Classe da montadora para o veículo:

Listagem 5: *Classe MontadoraVeiculoBuilder*

```
1 class MontadoraVeiculoBuilder implements MontadoraBuilder {
2     private Veiculo veiculo;
3     public void montar(char tipo, String modelo) {
4         switch (tipo) {
5             case 'F': veiculo = new Ferrari(modelo); break;
6             case 'J': veiculo = new Jeep(modelo); break;
7         }
8     }
9     public Veiculo getVeiculo() {
10         return veiculo;
11     }
12 }
```

12 }

Classe que ordena a construção do veículo:

Listagem 6: *Classe Montadora*

```
1 class Montadora {  
2     private MontadoraBuilder builder = new MontadoraVeiculoBuilder();  
3  
4     public Veiculo montar(char tipo, String modelo) {  
5         builder.montar(tipo, modelo);  
6         return builder.getVeiculo();  
7     }  
8 }
```

Classe com um exemplo de uso pelo cliente:

Listagem 7: *Classe Cliente*

```
1 public class Cliente {  
2     public static void main(String[] args) {  
3         new Cliente().executar();  
4     }  
5     public void executar() {  
6         Montadora montadora = new Montadora();  
7  
8         Veiculo carro1 = montadora.montar('F', "Testarossa");  
9         Veiculo carro2 = montadora.montar('J', "Renegade");  
10  
11         System.out.println(carro1);  
12         System.out.println(carro2);  
13     }  
14 }
```

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.