

Padrão Builder em Java

Autores: Tiago Ferraz de Novais
Renato Gomes Souza
Haniel Pereira Albino da Silva
Carlos Eduardo Silva Santos
Ênio Felipe Teixeira Soares

Agenda

O objetivo dessa apresentação é apresentar o padrão Builder como solução para construção de objetos complexos em Java.

1. Introdução:

Motivo do porque usar o Builder

2. Implementação:

Estrutura e exemplo do código

3. Vantagens:

Quais os benefícios do Builder

4. Considerações finais:

Exemplo de aplicação

5. Referências:

Material de pesquisa

1. Introdução



O Builder é uma maneira organizada de criar objetos quando temos muitas informações

A falta do padrão Builder em códigos podem levar a:

- Dificuldade de leitura e manutenção do código
- Falta de informação sobre cada campo
- Campos opcionais virando “null” ou “false” se não preenchidos

2. Implementação



A estrutura do Builder feita para esse projeto:

// Construtor público para facilitar o uso por iniciantes

```
public OrderBuilder() {}
```

// Define o cliente do pedido

```
public OrderBuilder setCustomer(Customer customer) {  
    if (customer == null) {  
        throw new IllegalArgumentException("Cliente não pode ser nulo!");  
    }  
    this.customer = customer;  
    return this;  
}
```

// Adiciona um item ao pedido

```
public OrderBuilder addItem(Menuitem menuitem, int quantidade) {  
    if (menuitem == null) {  
        throw new IllegalArgumentException("Item não pode ser nulo!");  
    }  
    if (quantidade <= 0) {  
        throw new IllegalArgumentException("Quantidade deve ser positiva!");  
    }  
    items.add(new OrderItem(null, menuitem, quantidade));  
    return this;  
}
```

2. Implementação



```
// Cria o pedido
public Order build() {
    if (customer == null) {
        throw new IllegalStateException("Cliente deve ser definido!");
    }
    if (items.isEmpty()) {
        throw new IllegalStateException("Adicione pelo menos um item!");
    }
    Order order = new Order(customer);
    for (OrderItem item : items) {
        order.addItem(item.getMenuItem(), item.getAmount());
    }
    order.setStatus(OrderStatus.ACCEPTED);
    return order;
}
}
```

3. Vantagens



As vantagens do Builder são:

- Código autoexplicativo
- Objeto final imutável e seguro
- Ordem dos parâmetros não importa
- Fácil de ler e manter
- Validação centralizada

4. Considerações finais



No fim, foi visto que o Builder:

- Se destaca em criação de objetos com muitos parâmetros
- Garante imutabilidade
- Simplifica a construção e entendimento do código

Claro que com suas desvantagens, sendo mais apropriado para objetos complexos onde a clareza e manutenção são importantes

5. Referências



Gamma et al. (1994) Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994) "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley.

Bloch (2008) Bloch, J. (2008) "Effective Java", Prentice Hall, 2nd edition.

Freeman et al. (2004) Freeman, E., Robson, E., Bates, B., and Sierra, K. (2004) "Head First Design Patterns", O'Reilly Media.