

Padrões de Projeto

Template Method

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação

Agenda

- 1 Introdução
- 2 Estrutura do Template Method
- 3 Consequências do uso do Template Method
- 4 Template Method vs Strategy
- 5 Aplicabilidade
- 6 Mau cheiro de código: Missing Template Method
- 7 Trabalho

Introdução

Padrão Template Method

Este padrão é aplicável quando se deseja definir um algoritmo geral, que estabelece uma série de passos para cumprir um requisito da aplicação.

Porém, seus passos podem variar e é desejável que a estrutura da implementação forneça uma forma para que eles sejam facilmente substituídos.

Estrutura do Template Method

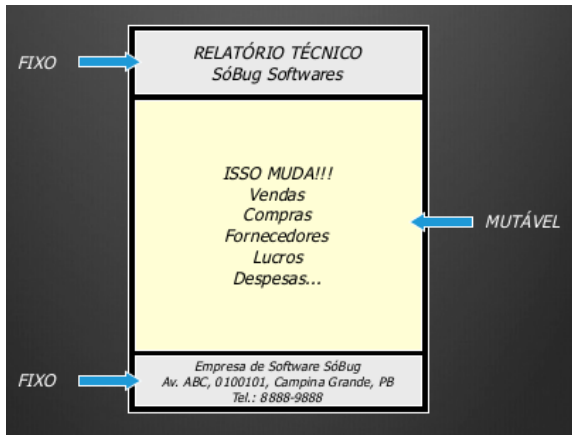
Para entender a estrutura do Template Method vamos utilizar como metáfora algo familiar a muitas pessoas: modelos de documento.

Um modelo de documento define algumas partes que são fixas e outras partes que devem ser introduzidas quando o documento propriamente dito for criado.

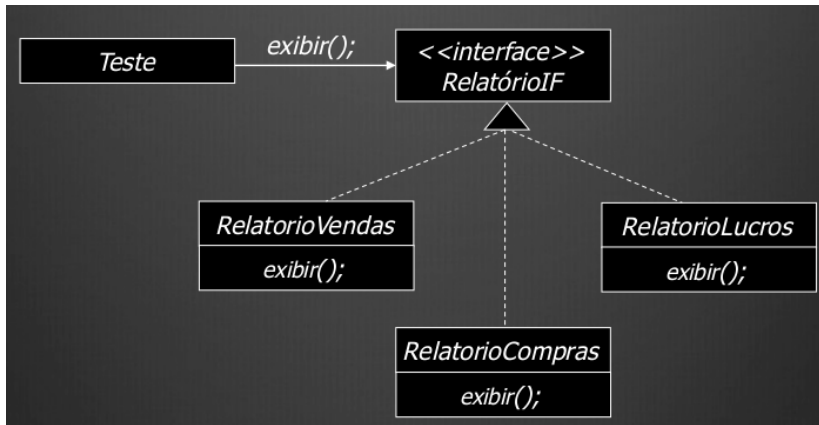
São lacunas que precisam ser completadas e que irão variar de documento para documento. O modelo provê uma estrutura que pode ser facilmente reaproveitada, simplificando sua criação.

Estrutura do Template Method

Como projetar???



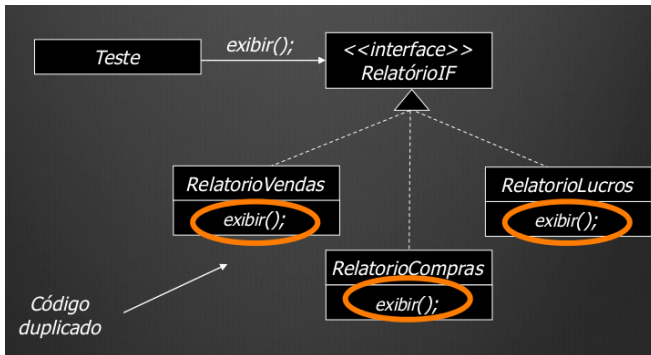
Estrutura do Template Method



Estrutura do Template Method

Problemas???

O cabeçalho e rodapé são iguais... não estou compartilhando código...e ainda estou duplicando



Estrutura do Template Method

Solução??? Template Method

Define-se um esqueleto do método `exibir..`

```
public void exibir() {  
    exibirCabecalho() ;  
    exibirConteudo() ;  
    exibirRodape() ;  
}
```

E define o comportamento MUTÁVEL como um método abstrato...

Estrutura do Template Method

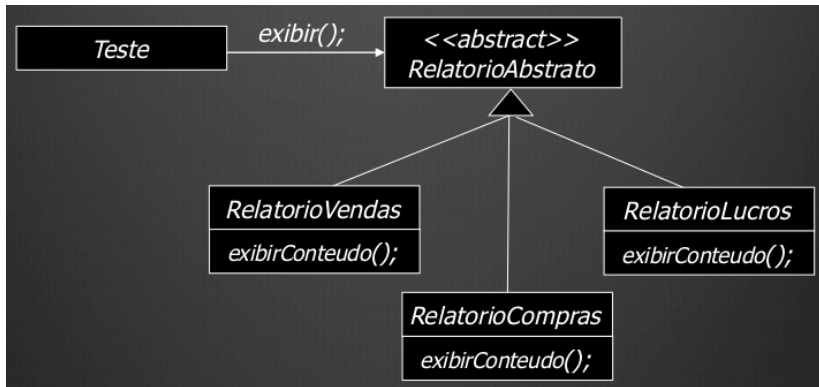
```
public void exibirCabecalho() {  
    System.out.println("Cabecalho");  
    System.out.println("...");  
    ...  
}
```

```
public void exibirRodape() {  
    System.out.println("Rodapé");  
    System.out.println("...");  
    ...  
}
```

```
public void abstract exibirConteudo();
```

Estrutura do Template Method

Projeto com o Template Method



Estrutura do Template Method

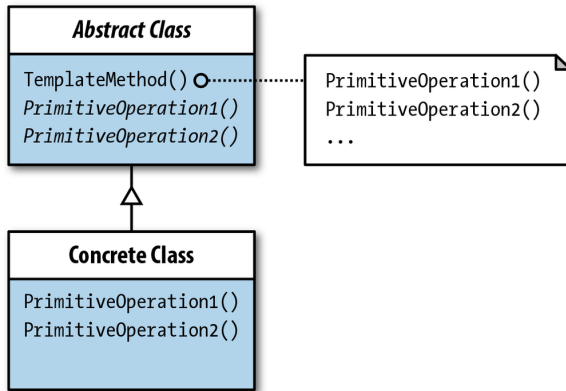
De forma similar, um Template Method é um modelo de algoritmo que possui algumas partes fixas e algumas partes variáveis.

As partes variáveis são lacunas que precisam ser completadas para que o algoritmo faça realmente sentido. As lacunas podem ser implementados nas subclasses.

- Caso seja uma lacuna obrigatória, o método deve ser definido como abstrato;
- Caso a implementação seja opcional, o método pode ser concreto e normalmente possui uma implementação vazia.

O algoritmo é representado através de um método na superclasse que coordena a execução.

Estrutura do Template Method



Estrutura do Template Method

AbstractClass

- Define as operações primitivas abstratas que subclasses concretas devem implementar.
- Implementa um método de molde (`templateMethod`) que define o esqueleto de um algoritmo.

ConcreteClass

- Implementa as operações primitivas para realizar etapas específicas de subclasse do algoritmo.
- Vale salientar que o método `templateMethod()` é **final**, ou seja, ele **não pode ser sobrescrito**.

ConcreteClasses esperam que **AbstractClass** implemente os passos invariantes do algoritmo.

Consequências do uso do Template Method

Com o Template Method podemos reaproveitar o código relativo à parte comum de um algoritmo, permitindo que cada passo variável possa ser definido na subclasse.

- Isso também é uma forma de permitir que a funcionalidade da classe que define o algoritmo básico seja estendida.

É importante lembrar que os modificadores adequados dos métodos devem ser utilizados para impedir que o contrato da superclasse com os seus clientes seja quebrado.

Consequências do uso do Template Method

Porém, **o uso da herança nesse padrão também traz algumas limitações:**

- ❶ A herança é **uma carta que só pode ser jogada uma vez**
 - Isso significa que uma classe que precise de comportamentos de duas outras classes só poderá fazer o uso da herança para uma delas.
- ❷ É que depois que uma implementação for instanciada não será mais possível alterar os passos do algoritmo. (Lembrar do Padrão Strategy)

Consequências do uso do Template Method

- Strategy é como Template Method exceto em sua granularidade. (dividido em mais partes - mais componentes)
- **Template Method** usa herança para variar parte de um algoritmo. **Strategy** usa delegação para variar o algoritmo inteiro.
- Strategy modifica a lógica de objetos individuais. Template Method modifica a lógica de uma classe inteira.

Aplicabilidade

Quando temos várias classes realizando basicamente a mesma tarefa com algumas pequenas modificações.

- Template de documentos
- Templates de projetos
- Template de configurações

Mau cheiro de código: Missing Template Method

Um mau cheiro, ou bad smell, se refere a algum indício de problemas no código que podem significar deficiências em sua modelagem.

Um mau cheiro é diferente de um bug, pois não necessariamente ele causa um erro na aplicação, mas traz outras consequências negativas como dificuldade de manutenção e falta de flexibilidade.

Um dos mau cheiros que são frequentemente detectados por essas ferramentas se chama Missing Template Method.

Mau cheiro de código: Missing Template Method

Missing Template Method

Ele se refere a dois componentes que possuem um número significativo de similaridades, mas não compartilham de uma mesma abstração, como uma interface, ou de uma mesma implementação, como um Template Method.

Ao se deparar com essa situação, é indicada a refatoração descrita.

Trabalho

Exportação de Dados

- Ler dados
- Formatar os dados
- Exportar

Descrição do Trabalho

Leia um arquivo .csv de cadastro de pessoas e exporte o mesmo para diferentes formatos:

- Ex: Texto Plano, Banco de Dados, PDF, HTML, XML, Planilha Eletrônica, Latex, SQL e etc
- Selecione no mínimo 4 formatos

Trabalho

Exemplo CSV

```
cpf;nome;idade;  
111.111.111-11;igor pereira;29;  
222.222.222.-22;betito;90;  
333.333.333-33;mifael;50;
```

Sugestão

- *parseDataAndGenerateOutput()* **TemplateMethod**
- *readData()* **PrimitiveOperation1()**
- *processData()* **PrimitiveOperation2()**
- *writeData()* **Método Concreto (AbstractClass)**

Padrões de Projeto

Template Method

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação