



Universidade Federal Rural do Semi-Árido
Departamento de Engenharias e Tecnologia
Projeto Detalhado de Software
Profa. Huliane Medeiros da Silva

Lista de Exercícios - Padrões de Projetos

1. Factory Method

- a. Considere a Cantina da UFERSA, onde vários sanduíches diferentes podem ser feitos. Um sanduíche básico contém: duas fatias de pão, uma fatia de queijo, uma fatia de presunto e salada (estrutura base de qualquer sanduíche). No entanto, existem variações do sanduíche básico de acordo com os diferentes tipos de ingredientes (veja quadro). Tendo o código para o sanduíche base, aplique o padrão de projeto Factory Method para que os sanduíches abaixo possam ser feitos pela Cantina.

Ingredientes		Sandwiches		
		Lanchonete CG	Lanchonete JP	Lanchonete RT
Pão	Integral	X		
	Francês		X	
	Bola			X
Queijo	Prato	X		
	Mussarela		X	
	Cheddar			X
Presunto	De Frango	X	X	
	De Peru			X
Salada	Com verdura		X	
	Sem verdura	X		X

2. Abstract Factory

- a. Refine o código elaborado para a questão 1 com o objetivo de aplicar o padrão de projeto Abstract Factory. Para tanto, faz-se necessário:
- Definir uma interface `SandwichesIngredientFactory` com os respectivos métodos de criação (Factory Methods);
 - Defina uma interface para cada tipo de produto: `PãoIF`, por exemplo.
 - Criar as fábricas concretas que devem implementar a interface definida.
 - `SandwichesIngredientFactoryCG`, `SandwichesIngredientFactoryJP` e `FabricaDeSandwichesRT`;
 - Os Factory Methods devem ser implementados nas respectivas fábricas;
 - Modifique o método para montagem do sanduíche, para que o mesmo passe a

- receber como parâmetro a fábrica específica.
- vi. Com as alterações sugeridas, a Cantina será a responsável por criar o sanduíche e terá, por composição, uma fábrica de sanduíches.
- vii. Modifique, ainda, a classe principal, main, para que as modificações sejam refletidas.

3. Factory Method:

- a. Explique porque o código abaixo não implementa o Factory Method Pattern. Modifique o código para que ele passe a implementar esse padrão.

```
public class XMLReaderFactory {  
    // Esse método retorna uma instância de uma classe // que implementa a interface XMLReader.  
    // A classe específica que ele cria e retorna  
    // é baseada em uma propriedade do sistema. public static XMLReader  
    createXMLReader();  
}  
  
public interface XMLReader {  
    public void setContentHandler(ContentHandler handler);  
    public void parse(InputStream is); }
```

4. Decorator:

- a. Ainda no contexto da Cantina da UFERSA, cada um dos lanches disponíveis no cardápio possui diferentes versões: normal, gourmet e vegano. Analise esse contexto e implemente o padrão Decorator, de forma que possa reaproveitar o máximo os métodos/atributos em comum entre cada um dos lanches (objetos). Dica: pode-se aplicar o Decorator, onde o lanche gourmet e vegano são “variantes” de um lanche comum.

5. Facade:

- a. Defina o padrão Facade.
- b. Quando o Facade deve ser usado?
- c. Como se dá a implementação deste padrão? Descreva a terminologia e estrutura (uml).

6. Observer:

- a. Dado o código abaixo de uma classe Subject (do padrão Observer):

```
interface Observer {  
    public void update(Subject s);  
}  
  
class Subject {  
  
    private List<Observer> observers=new ArrayList<Observer>();  
  
    public void addObserver(Observer observer) {  
        observers.add(observer);  
    }  
  
    public void notifyObservers() {  
        (A)  
    }  
}
```

Implemente o código de notifyObservers, comentado com um (A) acima.

7. Adapter:
 - a. Quando o padrão Adapter deve ser preferido sobre os outros padrões estruturais?
8. Composite:
 - a. Como o padrão Composite ajuda a consolidar a lógica condicional a nível de sistema. Em outras palavras, como esse padrão contribui para reduzir a necessidade de identificar a natureza de um componente em um sistema?
 - b. O padrão Composite pode ser usado se apenas poucos objetos têm filhos e quase todos os demais em uma coleção são objetos folha, isto é, sem filhos? Se positivo, como esse padrão seria implementado?
9. Template Method (vale 1 ponto extra):
 - a. Descreva um problema (diferente do que foi apresentado em sala e também diferente dos descritos na bibliografia base e complementar da disciplina) e proponha uma solução em que o padrão template method pode ser usado para resolver tal problema. Descreva todos os detalhes da solução (estrutura, diagrama, códigos, etc).