ESCOLA POLITÉCNICA

Disciplina: TECNOLOGIA DE CONSTRUÇÃO DE SOFTWARE (CMP 1069)

Prof. MSc. Anibal Santos Jukemura

**Implementação N2 – peso 2**

Segue contexto para implementação em JAVA:

Descrição da atividade:

Considere uma lanchonete conhecida por MacDinos. A MacDinos fornece sanduíches, como prato principal, batata-frita ou ovo frito como acompanhamento e refrigerantes dino-cola e dino-soda como bebidas.

Imagine uma implementação que utilize uma única instância de fábrica que produza três tipos de sanduíches: MacRaptor, MacRex e o MacMamute.

Imagine também uma instância única de uma fábrica que produza os acompanhamentos: MacBatata, MacOvoFrito.

Também existe uma fábrica como instância única que produz os refrigerantes DinoCola e DinoSoda.

As fábricas possuem um serviço de personalização, sendo que o cliente pode optar por acrescentar DinoBacon e/ou DinoMolho.

Imagine que é dia de promoção e a MacDinos criou combos especiais, que podem conter os sanduíches, os acompanhamentos, os refrigerantes e os itens extras personalizados, sendo tudo organizado de acordo com a vontade do cliente.

Diante do contexto apresentado, desenvolva uma aplicação em JAVA que permita a elaboração dos Combos. Considere os seguintes aspectos:

* Para a produção dos sanduíches, considere o uso do Padrão **Simple Fabric** com **Singleton**. A fábrica deve ser implementada em sua definição clássica do GoF, sendo que deverá ser instanciadas uma única vez (Singleton).
* Para a produção dos acompanhamentos, considere o uso do Padrão **Simple Fabric** com **Singleton**. A fábrica deve ser implementada em sua definição clássica do GoF, sendo que deverá ser instanciadas uma única vez (Singleton).
* Para a produção dos refrigerantes, considere o uso do Padrão **Simple Fabric** com **Singleton**. A fábrica deve ser implementada em sua definição clássica do GoF, sendo que deverá ser instanciadas uma única vez (Singleton).
* Utilize o Padrão **Decorator** para permitir personalizar os sanduíches com a DinoBacon ou com o DinoMolho.
* Utilize o Padrão **Composite** para elaborar os combos. Cada combo deve permitir a combinação dos sanduíches (personalizados ou não) com os acompanhamentos e com os refrigerantes, bem como com outros combos personalizados. Ou seja, um pedido pode vir com um ou mais combos um dentro do outro e mesclados com sanduíches e/ou refrigerantes.
* Os sanduíches, os acompanhamentos e os refrigerantes devem implementar uma mesma interface denominada Produto (que será usada para compor os combos). (Obs.: isso permite DinoCola com DinoMolho... kkk, mas podemos ignorar).
* O Padrão Decorator deve ter como atributo um Produto do tipo protected.
* A interface Produto deve ter um método pedido() que deverá ser implementado nas classes DinoBacon e DinoMolho, emitindo pelo menos uma mensagem que identifique qual personalização foi usada. O Mesmo vale para o acompanhamento, uma vez que essa classe também deve implementar um Produto. Considere declarar um método get() na interface Produto, caso julgue necessário.
* Para testar o programa, crie todas as fábricas necessárias crie alguns Combos. Seguem exemplos de saídas:

Tabela

Descrição gerada automaticamente com confiança média

Texto

Descrição gerada automaticamente com confiança média