

Atividade Externa da Disciplina (AED)

**INTRODUÇÃO: Composite** **Desing Pattern (Leia: Importante)**

O Composite é um padrão de projeto estrutural que permite que você componha objetos em estruturas de árvores e então trabalhe com essas estruturas como se elas fossem objetos individuais.

Usar o padrão Composite faz sentido apenas quando o modelo central de sua aplicação pode ser representada como uma árvore.

O maior benefício dessa abordagem é que você não precisa se preocupar sobre as classes concretas dos objetos que compõem essa árvore. Você não precisa saber se um objeto é um produto simples ou uma caixa sofisticada. Você pode tratar todos eles com a mesma interface. Quando você chama um método os próprios objetos passam o pedido pela árvore.

Segue diagrama clássico:

Diagrama

Descrição gerada automaticamente

Figura . Composite Pattern Design. (Fonte: autor)

**1. Objetivos da Atividade**

* Aplicar o padrão de projeto COMPOSITE na construção de um sistema;
* Utilizar o padrão construcional FABRIC METHOD para auxiliar na criação dos objetos;
* Empregar, adequadamente, os conceitos de orientação a objetos utilizando uma linguagem de programação

**2. Descrição da Atividade**

O aluno deverá definir um sistema e aplicar o conceito do COMPOSITE *Design Pattern* aprendido na construção deste sistema.

A atividades principal consiste em desenvolver uma aplicação em Java que use o padrão de projeto COMPOSITE na construção do sistema**.**

**SISTEMA 1** Considere um projeto em Java chamado ProjetoFabricaComposite que simule uma Micro Empresa que fornece Cestas de Café da Manhã.

Para que a empresa funcione, imagine que ela possua duas fábricas principais:

* Uma fábrica de Salgados que produza coxinha, quibe e esfirra
* Uma fábrica de Doces que produza brigadeiro, bolo e sorvete.

Essa micro empresa monta três tipos de cestas de café da manhã:

* Cesta de salgados: que pode ter quaisquer combinações dos salgados supracitados.
* Cesta de doces: que pode ter quaisquer combinações dos doces supracitados.
* Cesta mista: composta por salgados e doces.

Implemente uma aplicação Java que represente as duas fábricas: uma de salgados e outra de doces e, através do padrão Composite, represente formas em que seja possível implementar qualquer uma das configurações de cestas de café da manhã possíveis.

Segue exemplo de saída:

Tabela

Descrição gerada automaticamente com confiança média

Dicas e obrigatoriedades: implemente uma interface (Produto) com um método **listar** e **getTipo**. Tanto a fábrica **abstrata** de salgados, quanto a fábrica **abstrata** de doces implementarão essa interface e devem ter ambas um atributo private string tipo que vai guardar o tipo do produto fabricado: coxinha, bolo, esfirra, brigadeiro, etc. As implementações concretas de coxinha, quibe, esfirra, bolo, brigadeiro e sorvete se encarregarão de alterar seus tipos (atributo da classe abstrata produto). A Classe composite (pode se chamar CaixaCafeManha), vai implementar um ArrayList de produtos, sejam da fábrica de salgados, seja da fábrica de doces ou a própria “caixa” instanciada da classe CaixaCafeManha. Lembrando que essa caixa CaixaCafeManha também tem nome.

A composição 3 deve ser uma Caixa de produtos da fábrica de salgados ou uma Caixa de produtos fábrica de doces.

**3. Cronograma**

* Serão reservados os 10 minutos finais de cada aula, a partir de 20/05/2023, para orientação da produção da AED;
* A data final de entrega é 09/06/2023.

**4. Forma de Registro**

Apresentação e entrega de um arquivo em ZIP contendo o programa desenvolvido em JAVA.

**5. Critérios de Avaliação**

Como descrito no item Avaliação, o valor da atividade AED é de 1,0 ponto, que será computado no valor das atividades que comporão a nota N2.  Serão também computadas 12 h.a.

**6.** **Bibliografia de Consulta**

ZIVIANI, Nivio. **Projeto de algoritmos com implementações em Java e C++.** São Paulo: Cengage Learning, 2006.

**7. Bibliografia Complementar**

SZWARCFITER, Jayme L. ***Estruturas de dados e seus algoritmos****.* 3. ed. Rio de Janeiro: LTC, 2010

REFACTORING GURU. Disponível em: <https://refactoring.guru/pt-br/design-patterns/composite> . Acessado em maio de 2023.