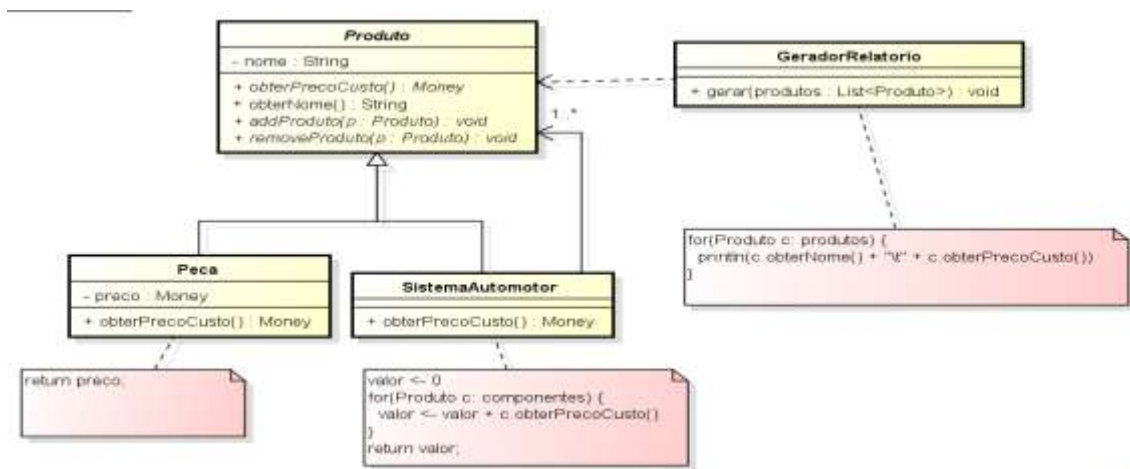


Informações preliminares

Esse trabalho envolve os padrões do catálogo Gang of Four (GoF), objeto de estudo em nossas aulas no 2ª bimestre. Neste trabalho, você deve analisar cada uma das situações-problema apresentadas. Para cada uma delas, você deve construir uma solução de projeto que use algum dos padrões GoF. Para apresentar a sua solução para cada situação-problema, você deve construir um diagrama de classes que especifique os componentes da solução de forma detalhada. Você deve identificar qual padrão foi utilizado para cada situação-problema.

Forneça em forma de nota explicativa no diagrama de classes de cada solução um esboço da implementação (em linguagem Java) do corpo dos métodos necessários para implementar cada padrão. Como exemplo do nível de detalhamento esperado no trabalho em cada situação problema, considere o diagrama de classes abaixo, apresentado em aula no contexto do padrão Composite.



Todas as situações problema apresentadas neste trabalho estão no contexto de uma aplicação corporativa para uma loja virtual. Esse mesmo contexto já foi utilizado no trabalho prático do 1º bimestre e envolve classes de domínio como Venda, ItemVenda, Produto, Cliente e Pagamento.

Situações-problema

1. Considere que empresa que possui a loja virtual adquiriu uma base de clientes de outra empresa. Considere ainda que os clientes da nova base devam ser integrados à base atual de clientes. Esse clientes da nova base são representados pela classe externa *ClienteX* que deve ser utilizada durante a integração, embora não possa ter seu código fonte alterado. Existe a restrição de que a região de código que utiliza a classe *Cliente* já existente não deve ser alterada. Sua solução de projeto deve ser tal que permita que a loja virtual incorpore a nova base de clientes.
2. Considere que os comandos (requisições) que a aplicação Web (correspondente à loja virtual) receberá do cliente (navegador) serão passados para ela por meio de um parâmetro do tipo *String* pelo *request*. Considere ainda que essa aplicação deve Interpretar essa string na camada de apresentação identificando o comando específico que a aplicação deve realizar e despachando o controle para uma classe de tratamento de cada comando.
3. Considere que os itens de uma venda previamente realizada por um cliente devem ser separados para a entrega a esse cliente. Estes itens estão localizados fisicamente em diversos armazéns, a diferentes distâncias do local de entrega. Forneça uma solução de projeto para permitir a notificação aos armazéns para que estes façam a separação de um item para entrega. Os armazéns são notificados em uma fila sequencial por ordem de crescente de distância do local de entrega, sendo consultados nesta ordem. Se o armazém consultado tiver o item em estoque, deve enviá-lo ao cliente. Em caso contrário, a requisição é repassada para o próximo armazém na fila.
4. O catálogo de produtos da loja virtual é uma estrutura hierárquica de *categorias de produtos* e de *produtos*. Cada categoria pode conter diversas outras categorias e produtos. Não há um limite para os níveis de hierarquia do catálogo. Forneça uma solução de projeto que represente essa estrutura correspondente ao catálogo de produtos da loja virtual. Existe uma região de código que deve produzir uma listagem (relatório) em que cada linha apresenta o nome de uma categoria ou produto, e a respectiva quantidade em estoque. Para uma categoria, a quantidade em estoque deve ser o somatório das quantidades em estoque de seus componentes.
5. Um produto disponível para venda na loja virtual pode ser encomendado com alguns adicionais que podem ser escolhidos pelo cliente no momento da realização da venda. Cada adicional selecionado possui um valor, que será acrescido ao valor da venda. Por exemplo, o cliente pode solicitar que o produto seja embalado para presente, ou pode ser incluída uma proteção antichoque para sua entrega. O método que calcula o valor total da venda não deve depender destas diversas variações que podem ser adicionadas ao produto. Modele uma solução de projeto que permita esta flexibilidade na combinação de adicionais do produto e o cálculo do total da venda.
6. No contexto da arquitetura de uma aplicação corporativa vimos o conceito de operação de sistema. Estas operações correspondem a uma interface de programação que simplifica o acesso à lógica de domínio. Suponha que parte das funcionalidades da loja virtual possa

ser acessada por meio de um aplicativo móvel, a ser desenvolvido. Esse aplicativo precisa de uma interface mais simples para acesso aos serviços da loja virtual. Mais especificamente, deve ser criada uma classe na camada de serviço que defina as operações de sistema para um caso de uso de realizar uma compra por meio do aplicativo móvel.

7. Na loja virtual, deve ser implementado um mecanismo de *logging*, que pode posteriormente ser usado para depuração da aplicação ou para auditoria. Deve ser criada uma classe para permitir o acesso (leitura e escrita) ao arquivo de log. Essa classe deve ter somente uma instância globalmente acessível por toda a aplicação.
8. Forneça uma solução de projeto para percorrer e imprimir o catálogo de produtos da loja virtual de forma sequencial, sem expor a estrutura interna do catálogo. A região de código que precisar desse serviço deve atuar solicitando sempre o próximo elemento do catálogo, podendo assim percorrer todos eles de forma transparente.
9. Suponha que a aplicação correspondente à loja virtual utilize o padrão DAO para implementar o acesso ao mecanismo de armazenamento persistente. Suponha ainda que devemos criar um tipo ProdutoDAO, que pode possuir diversas implementações. Por exemplo, pode haver uma implementação que permita realiza a persistência em um banco de dados relacional, e outra que permita a persistência em um arquivo XML. Forneça uma solução de projeto que permita criar um objeto que implemente a interface do tipo ProdutoDAO, com restrição de que o cliente do objeto a ser criado não conheça a solução específica de persistência a ser instanciada.
10. Considerando a situação problema do exercício anterior. Suponha que agora devem existir diferentes DAOs, não somente para a classe Produto. Por exemplo, devem ser criados DAOs para as classes Venda e Cliente. A aplicação deve selecionar a solução específica de persistência, e todos os DAOs devem seguir a mesma solução, com restrição de que o cliente dos DAOs a serem criados não conheça a solução específica de persistência usada.
11. Apresente uma solução de projeto para que a aplicação da loja virtual tenha um algoritmo padrão que realiza o cálculo do imposto total sobre o valor de uma venda. Este algoritmo consiste de uma sequência de passos para o cálculo de cada imposto. Entretanto alguns destes passos são dependentes de certas condições. O algoritmo padrão deve ser independente da fórmula de cálculo de cada imposto específico.
12. Apresente uma solução de projeto para que os métodos da classe Venda respondam de forma adequada dependendo do estado da venda. Alguns métodos desta classe são dependentes do estado. Por exemplo, os métodos realizarEntrega e registrarPagamento possuem diferentes comportamentos em função da venda estar quitada ou aberta.
13. Pode haver diferentes estratégias (algoritmos) intercambiáveis para calcular o desconto de uma venda. Faça com que o cliente possa aderir a uma das estratégias de desconto dinamicamente. Por exemplo, o desconto pode ser calculado utilizando um percentual do valor total da venda, ou, em função da natureza dos produtos adquiridos em uma venda.

Procedimentos para entrega do trabalho

O que entregar: crie um arquivo compactado de nome SEU_NOME_COMPLETO.zip. Esse arquivo deve conter os seguintes itens:

- Arquivo do Astah (<http://astah.net/>) com as soluções de projeto. Crie diversos diagramas de classe, uma para cada solução projeto. Para cada diagrama de classes criado, seu nome deve ser igual ao padrão GoF utilizado na solução.
- Arquivo (em formato PDF) contendo as impressões de todos os diagramas de classes contidos no arquivo Astah.

Quando entregar: veja na página do curso.