

Disciplina: Padrões de Projetos

Atividade de Revisão

Nome: Francisco Kleber Souza Oliveira

1. Qual é a principal intenção do Padrão Adapter (Adaptador)? Apresente um cenário de desenvolvimento em Java onde a aplicação deste padrão seria a solução mais eficiente, focando em como ele resolve o problema de incompatibilidade.

A principal intenção do padrão Adapter é permitir a compatibilidade entre classes, que seriam incompatíveis devido às suas interfaces. Isto quer dizer que um cliente existente pode utilizar uma nova funcionalidade que possui uma interface diferente. O sistema de som de carro é um cenário aonde o Adapter seria eficiente pois o cliente que já possui uma lógica existente e está configurado para tocar apenas rádio. O padrão Adapter é implementado utilizando uma classe intermediária. Quando um cliente inicia uma chamada o Adapter intercepta essa chamada, internamente a traduz para um método que o novo componente entende.

2. O Padrão Adapter pode ser implementado de duas formas principais: como Adapter de Classe ou como Adapter de Objeto. Explique a diferença estrutural entre essas duas abordagens e discuta qual delas é geralmente mais flexível em Java e por quê.

O Adapter de Classe herda simultaneamente da classe adaptee e da interface target, já o Adapter de Objeto o Adapter é uma classe que implementa a interface target e contém uma instância da classe Adaptee. O Adapter de objeto é mais flexível, devido à capacidade de adaptar subclasses graças ao polimorfismo.

3. Qual é a principal motivação para usar o Padrão Factory Method, em termos de programação orientada a objetos (POO)? Explique como ele contribui para a observância do Princípio Aberto/Fechado (OCP) do SOLID.

A principal razão é separar a lógica de criação desses objetos da lógica de uso desse objeto. Em relação ao OCP, ele permite a criação de novos produtos, você pode adicionar um novo produto sem alterar o código existente, o cliente interage apenas com a interface ou classe do produto e o creator, ou seja, não exige a modificação do código.

4. Descreva os componentes essenciais do Padrão Factory Method (Produto, Criador e Fábrica Concreta) e explique a responsabilidade principal de cada um. Como o Padrão garante que a criação de objetos seja deferida para as subclasses?

O primeiro é o product que visa estabelecer um contrato comum de objetos a serem criados, concrete product implementar os métodos definidos na interface Product, Creator definir o esqueleto da lógica e declarar o factory method e Concrete Creator implementar o factory method para retorna um Concret Product. Ele garante que a criação seja deferida através do polimorfismo e da inversão de controle.

5. Qual o significado da sigla GRASP? Em poucas palavras, qual é o objetivo fundamental desses padrões no projeto de software orientado a objetos, e como eles se diferenciam dos Padrões de Projeto GoF (Gang of Four)?

A principal função do padrão GRASP é ajudar os desenvolvedores a tomarem boas decisões estruturais e design durante a construção de um sistema. O padrão GOF é visto mais como uma solução de problema já o GRASP é uma ferramenta conceitual que você usa como sua classe devem ser projetadas e quais responsabilidades.

6. Como os padrões GRASP são classificados? Descreva quais padrões fazem parte de cada classificação.

O padrão GRASP é classificado em Estruturação que define qualidade interna, criação definem o fluxo de objetos e interação e suporte são usados para refinar o objeto para maior robustez e adaptabilidade.

7. O Princípio da Informação Especializada (ou Information Expert) é um dos Padrões GRASP (Padrões Gerais de Atribuição de Responsabilidade de Software). Ele é considerado um dos padrões fundamentais e mais importantes do GRASP. Explique sua definição, propósito e qual problema ele visa resolver e como.

Ele define uma responsabilidade a uma classe que possui informação necessária para cumpri-la, ele garante que os métodos e os dados relacionados estejam encapsulados na mesma classe, os maiores problemas são baixo coesão e alta acoplamento, ele resolve esses problemas por meio do encapsulamento ao colocar método na classe que detém os dados e alta coesão pois todas as suas responsabilidades estão diretamente relacionadas aos dados que ela contém e gerencia.

8. O Information Expert é frequentemente chamado de padrão fundamental mais importante do GRASP. Qual é a regra de atribuição de responsabilidade definida por este padrão? Apresente um exemplo em que o Information Expert seria aplicado para determinar qual classe deve ter a responsabilidade de calcular o preço total de um carrinho de compras.

Atribuir uma responsabilidade à classe que possui a informação necessária para cumprir essa responsabilidade. Como a responsabilidade é o preço total do carrinho, o cálculo deve ser distribuído, a responsabilidade principal deve ser atribuída ao Carrinhodecompras é a principal especialista.

9. Qual é a regra básica do Padrão Criador (Creator) do padrão GRASP? Explique como esta regra visa determinar qual classe é a "melhor" responsável por criar uma instância de uma dada classe, em vez de simplesmente delegar a criação a uma fábrica genérica.

Atribua a responsabilidade de instancia de objetos A à objetos B se B for agregador. Ele identifica a classe que se beneficia ou é essencial para a existência do objeto criado. O PADRÃO CRIADOR reduz o acoplamento ao evitar que uma classe alectória chame o construtor.

10. O Padrão Criador lista quatro condições principais que indicam que a Classe B deve ser responsável por criar instâncias da Classe A. Liste e descreva brevemente pelo menos três dessas condições, concentrando-se nos relacionamentos de classes (agregação/composição e associação).

Agregação aonde a Classe B agrega ou contém instâncias da Classe A. Associação define a Classe B usa intimamente a Classe A. Registro inseri a Classe B registra ou mantém um registro das instâncias da Classe A.

**11. Quais são os principais benefícios de design alcançados ao seguir o Padrão Criador? Em particular, como o uso deste padrão ajuda a manter o Baixo Acoplamento no sistema?**

Alta coesão e Baixo acoplamento, Concentração de dependência ao atribuir a responsabilidade de criação à Classe B, a dependência da classe concreta A, evitando fábricas genéricas como utilizando classes que já existem no modelo do domínio, não adicionando novas classes ao sistema apenas para a criação.

**12. Considere um cenário em que a classe Venda precise criar um objeto ItemVenda. Aplique o princípio do Criador e o princípio do Especialista na Informação para justificar por que a classe Venda é a candidata ideal para criar o ItemVenda.**

Pois a classe Venda agrega e contém o itemvenda por ser a ordem natural, o especialista atribui a responsabilidade de criação a classe Venda, pois ele possui informação e responsabilidade sobre a nova instância.