

Java - Padrões de Projetos Comportamentais

Questão 1: Sobre o padrão de projetos Chain of Responsibility responda:

- Para que serve o padrão de projetos Chain of Responsibility ?

R: Formar uma cadeia de objetos receptores e passar uma requisição pela mesma, dando a chance a mais de um objeto a responder a requisição ou colaborar de alguma forma na resposta.

- Quando devemos usar o padrão de projetos Chain of Responsibility

R: Quando mais de um objeto pode responder a uma requisição e não se sabe qual a prioritário, não se quer especificar o receptor explicitamente, estes objetos são especificados dinamicamente.

- Cite as vantagens e desvantagens do padrão de projetos Chain of Responsibility

R: Vantagem: Não se sabe a classe ou estrutura interna dos participantes. Pode usar Mediador para desacoplar ainda mais. flexível, em tempo de execução.

- Explique o código apresentado na Aula Prática - 13 Padrões de Comportamento - Chain of Responsibility

- Explique a classe abstrata utilizada

R: Esta classe contém a ordem do log e onde pega o próximo e imprime a mensagem

- Explique todas as classes implementadas

R: É criada class para cada objeto no exemplo passado vc cria class ConsoleLogger, e e no main você cria a ordem log->filler;

Qual a classe mais importante que define um Chain of Responsibility? Porque?

R: AbstractLogger.

Questão 2: Sobre o padrão de projetos Command responda:

- Para que serve o padrão de projetos Command?

R: Para encapsular uma requisição como um objeto, permitindo parametrização, enfileiramento, suporte a histórico.

- Quando devemos usar o padrão de projetos Command

R: Quiser parametrizar ações genéricas, Quiser enfileirar e executar comandos de forma assíncrona, em outro momento; Quiser permitir o undo de operações, dando suporte a históricos;

- Cite as vantagens e desvantagens do padrão de projetos Command

R: Vantagens: Objeto que evoca a operação e o que executa são desacoplados, Pode ser usado junto com Composite para formar comandos complexos;

- Explique o código apresentado na Aula Prática - 14 Padrões de Comportamento - Command

- Explique a interface utilizada

R: A interface coloca as regras de executar e undo.

- Explique todas as classes implementadas

R: A classe implementadas pela class order oque e onde tem as regras, depois cria a classe stock que irar solicitar

- Qual a classe mais importante que define um Command? Porque?

R: Borker, e a classe que solicita os comandos

Questão 3: Sobre o padrão de projetos Interpreter responda:

- Para que serve o padrão de projetos Interpreter?

R: Para definir a gramática de uma linguagem e criar um interpretador que leia instruções nesta linguagem e interprete-as para realizar tarefas.

- Quando devemos usar o padrão de projetos Interpreter

R: Existe uma linguagem a ser interpretada que pode ser descrita como uma árvore sintática;

- Cite as vantagens e desvantagens do padrão de projetos Interpreter

R: É fácil mudar e estender a gramática, Gramáticas complicadas dificultam.

- Explique o código apresentado na Aula Prática - 15 Padrões de Comportamento - Interpreter

- Explique a interface utilizada

R: Classe interface onde tem as regras para receber os dados;

- Explique todas as classes implementadas

R: As classe terminalExpression onde e feita a interpretar os dados string as classe orExpression e arExpression onde e feita a comparacao se e maior ou menor.

- Qual a classe mais importante que define um Interpreter? Porque?

R: TerminalExpression, onde e feita a interpretação dos dados

Questão 4: Sobre o padrão de projetos Iterator responda:

- Para que serve o padrão de projetos Iterator?

R: Prover uma forma de acessar os elementos de um conjunto em sequência sem expor a representação interna deste conjunto.

- Quando devemos usar o padrão de projetos Iterator

R: Quiser acessar objetos agregados (coleções) sem expor a estrutura interna, quiser prover diferentes meios de acessar tais objetos, quiser especificar uma interface única e uniforme para este acesso.

- Cite as vantagens e desvantagens do padrão de projetos Iterator

R: Múltiplas formas de acesso, Interface simplificada.

- Explique o código apresentado na Aula Prática - 16 Padrões de Comportamento - Iterator

- Explique as interfaces utilizadas

R: Esta e as classes interface iterator, Container .

- Explique todas as classes implementadas

- Qual a classe mais importante que define um Iterator? Porque?

R: NameRepository

Questão 5: Implemente os problemas abaixo utilizando seus conhecimentos sobre os padrões de projetos Comportamentais:

- Implemente uma solução em Java utilizando o padrão de projetos Chain of Responsibility para representar uma cadeia alimentar. Exemplo: Vegetal -> Grilo -> Sapo -> Cobra -> Gavião
- Implemente uma solução em Java utilizando o padrão de projetos Command para representar um Interruptor. Este interruptor deve ser capaz de funcionar para qualquer tipo de circuito, motor que for implementado através de uma interface genérica que represente este interruptor.
- Implemente uma solução em Java utilizando o padrão de projetos Interpreter para representar um Interpretador de formas. Este interpretador deverá interpretar qual a forma (Quadrado, Retângulo ou Círculo), de acordo com os parâmetros que o usuário fornecer.
- Crie uma adaptação ao exemplo Aula Prática - 16 Padrões de Comportamento - Iterator para suportar novos Iterators de Idade e Cpf para os nomes relacionados no exemplo.